Tiedostomuotojen piirteet pitkäaikaissäilytyksen kannalta

Janne Pulkkinen

Pro gradu
HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Helsinki, 5. toukokuuta 2019

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Laitos — Institution — Department
Matemaattis-luonnontieteellinen	Tietojenkäsittelytieteen laitos

Tekijä — Författare — Author
Janne Pulkkinen

Työn nimi — Arbetets titel — Title

Tiedostomuotojen piirteet pitkäaikaissäilytyksen kannalta

Oppiaine — Läroämne — Subject
Tietojenkäsittelytiede

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Pro gradu	5. toukokuuta 2019	63

Tiivistelmä — Referat — Abstract

Pitkäaikaissäilytyksellä pyritään pitämään erilaisia aineistoja kuten kuvia, asiakirjoja ja elokuvia käyttökelpoisina pitkän aikaa aineiston julkaisun jälkeenkin. Tiedostomuotojen vanheneminen, puutteelliset kuvaavat tiedot ja aineiston häviäminen vaarantavat aineistojen käyttökelpoisuuden tuleville sukupolville. Nämä vaaratilanteet pyritään estämään pitkäaikaissäilytyspalveluja käyttäen, joissa aineistot ja niitä koskevat metatiedot kerätään ja säilytetään kymmenien tai jopa sadan vuoden ajan. Palvelun vastuulla on tälloin pitää aineisto käyttökelpoisena niin pitkään, kun aineisto on säilytyspalvelun säilytyksessä.

Koska aineiston käyttökelpoisuus pitää taata hyvin pitkäksi aikaa, tärkeäksi kysymykseksi muodostuu säilytykseen valittavat tiedostomuodot. Tämä tutkielma analysoi käytössä olevia tiedostomuotoja ja niistä löydettyjä piirteitä, jotka tekevät tiedostomuodoista otollisia pitkäaikaissäilyttämistä varten. Näitä piirteitä hyödyntäen luotiin tarkistuslista, jonka avulla voidaan arvioida tiedostomuodon kelvollisuutta pitkäaikaissäilytystä varten. Luotu tarkistuslista arvioitiin Suomen kansallisen pitkäaikaissäilytyspalvelun kehittäjiltä saadun palautteen perusteella. Saadun palautteen perusteella tarkistuslistasta luotiin lopullinen versio, joka soveltuu tiedostomuotojen arviointiin.

Avainsanat — Nyckelord — Keywords
pitkäaikaissäilytys, tiedostomuodot

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Sisältö

1 Johdanto

Digitaalinen pitkäaikaissäilytys tarkoittaa informaation, kuten kulttuurillisten aineistojen ja asiakirjojen säilyttämistä kymmenien tai jopa sadan vuoden ajan. Pitkäaikaissäilytyksessä ei keskitytä pelkästään itse datan säilyttämiseen vaan myös sen käyttökelpoisuuteen, jolla pyritään säilyttämään materiaali myös tuleville sukupolville. Käyttökelpoisuuden säilyttäminen voi edellyttää materiaalin tiedostomuodon muuntamista toiseen esimerkiksi, kun alkuperäinen tiedostomuoto ei ole enää helposti käytettävissä ohjelmistojen vanhentuessa.

Suomessa kansallisesta pitkäaikaissäilytyksestä vastaa *CSC - Tieteen tietotekniikan keskus Oy*:n kehittämä pitkäaikaissäilytyspalvelu. Palvelun yhteistyöryhmään kuuluu arkistoja ja museoita ja palvelu itse keskittyy kulttuuriperintö- ja tutkimusaineistojen säilyttämiseen.

Tämä pro gradu -tutkielma käsittelee pitkäaikaissäilytystä esitellen ensin sen perusperiaatteet ja käytännön vaiheet sekä pitkäaikaissäilytyspalvelua tarjoavan tahon että pitkäaikaissäilytykseen materiaalia luovuttavan tahon osalta. Tutkielmassa paneudutaan mahdollisiin virhetilanteisiin, jotka vaarantavat materiaalin säilyvyyden tai ymmärrettävyyden, ja havainnollistetaan virhetilanteen syitä sekä mahdollisia keinoja, joilla nämä virhetilanteet on vältettävissä. Nämä virhetilanteet voivat koskea esimerkiksi puuttuvia metatietoja, metatietojen ristiriitaisuutta tai säilytetyn datan lukukelpoisuutta.

Tutkielman lopputuloksena luodaan tarkistuslista, jota voi käyttää mahdollisesti säilytykseen hyväksyttävien tiedostomuotojen arviointiin. Tarkistuslistan tavoitteena on olla hyödyllinen työkalu esimerkiksi pitkäaikaissäilytyspalvelua suunnittelevalle taholle sekä pitkäaikaissäilytykseen aineistoa lähettävälle taholle.

Tarkistuslistan eri vaiheet ja niiden tärkeysjärjestys selvitetään analysoimalla pitkäaikaissäilytystä koskevia tieteellisiä julkaisuja sekä analysoimalla käytössä olevia pitkäaikaissäilytysohjelmistoja, joiden lähdekoodi ja kehityshistoria on vapaasti saatavilla.

Luodun tarkistuslistan pätevyys arvioidaan keräämällä tarkastuslistasta palautetta alan asiantuntijoilta. Lisäksi tarkistuslistan käyttöä havainnollistetaan käyttäen esimerkkinä kahta laajassa käytössä olevaa tiedostomuotoa.

Tutkielma keskittyy ensisijaisesti digitaaliseen pitkäaikaissäilytykseen ja niissä käytettyihin tiedostomuotoihin. Digitaalisen pitkäaikaissäilytyksen

muihin osa-alueihin kuuluu muun muassa fyysisen materiaalin muuttaminen digitaaliseen muotoon ja materiaalin käyttökelpoisuuden ylläpitäminen emuloimalla vanhoja järjestelmiä. Emulointi pyrkii säilyttämään aineiston alkuperäisen suoritusympäristön sen sijaan, että aineiston käytettävyys säilytetään myös uudemmissa käyttöympäristöissä, jotka sisältävät huomattavasti erilaisen ohjelmistokokoelman. Esimerkiksi ohjelmistojen, interaktiivisten taideteosten ja muiden monimutkaisten aineistojen tapauksessa emulointi voi olla johdonmukaisempi säilytysmenetelmä [ERLS16]. Näitä aihealueita ei käsitellä tarkasti tutkielmassa, vaan tutkielman painopiste on tiedostomuodoissa ja niiden säilyttämiseen vaikuttavissa ongelmissa.

Tutkielman luvussa 2 käsitellään tutkielman aihealuetta: pitkäaikaissäilytystä ja sitä koskevia tärkeitä käsitteitä ja standardeja. Näiden termien selostamisen ohella lukijalle luodaan kuva pitkäaikaissäilytyspalvelun eri osa-alueista ja niiden toiminnasta. Luvun päätteeksi lukijalla on yleisluontoinen käsitys aineiston säilyttämiseen kuuluvista työvaiheista alkaen aineiston lähettämisestä ja päättyen aineiston hakemiseen takaisin palvelusta. Luvussa 3 käsitellään pitkäaikaissäilytystä haittaavia ongelmatilanteita, joissa säilytys ei toteudu aiemmin esitellyn ihanteellisen mallin mukaisesti. Pahimmassa tapauksessa nämä tilanteet voivat johtaa aineiston ymmärrettävyyden menettämiseen. Luvussa 4 käsitellään tiedostomuotoja keskittyen muutamiin asiakirja-, kuva- ja äänitiedostomuotoihin. Näistä tiedostomuodoista etsitään yhteisiä sekä erillisiä piirteitä. Lisäksi näitä piirteitä vertaillaan pitkäaikaissäilytyksen kannalta: mitkä piirteet tekevät tiedostomuodosta paremmin pitkäaikaissäilytettävän? Luvussa 5 käsitellään ohjelmistoja, joilla pitkäaikaissäilytyspalvelut voivat hoitaa tiedostomuotojen käsittelyn. Näihin toimenpiteisiin kuuluu muun muassa tiedostojen validointi, metatiedon kerääminen aineistosta sekä tarvittaessa tiedostomuodon muuttaminen toiseen. Luvussa 6 luodaan tiedostomuodoista ja ohjelmistoista löytyneiden seikkojen perusteella tarkistuslista, jolla voidaan arvioida tiedostomuodon sopivuutta pitkäaikaissäilytykseen. Tämä tarkistuslista arvioidaan käyttämällä sitä muutaman tiedostomuodon arviointiin sekä keräämällä palautetta pitkäaikaissäilytyksen asiantuntijoilta. Tutkielman päättävässä luvussa 7 selostetaan tutkielman johtopäätökset sekä listataan muita tutkimuskysymyksiä ja -aiheita, jotka tulivat esille tutkielman luomisen aikana.

2 Pitkäaikaissäilytys

Digitaalinen pitkäaikaissäilytys tarkoittaa materiaalin säilyttämistä digitaalisessa muodossa useiden kymmenien vuosien tai jopa satojen vuosien ajan. Pitkäaikaissäilytettyihin aineistoihin voi kuulua muun muassa asiakirjoja, kuvia, eläviä kuvia sekä äänitteitä. Koska pitkäaikaissäilytyksessä pyritään säilyttämään aineiston ymmärrettävyys, pelkästään tiedoston "bittien" säilyttäminen ei riitä. Esimerkiksi elokuvan tapauksessa voidaan säilyttää itse videotiedoston lisäksi tietoja elokuvan tuotannosta ja jakelusta. Elokuvan julkaisuvuosi, tuotantohenkilöstö ja elokuvan omistajat ovat esimerkkejä metatiedoista, jotka auttavat aineiston ymmärrettävyyden säilyttämisessä, mihin pelkkä videotiedoston säilyttäminen ei riitä.

Säilytetyn aineiston tulee pysyä käyttökelpoisena tuleville sukupolville myös tiedostomuotojen, ohjelmistojen ja/tai laitteiston vanhentuessa. Tämä voi edellyttää pitkäaikaissäilytetyn materiaalin muuntamista tarpeen tullen toiseen tiedostomuotoon, mikä varmistaa aineiston käyttökelpoisuuden jatkossakin.

Tässä tutkielmassa käsitellään suunnitelmallista, organisaatioiden tai valtioiden toteuttamaa pitkäaikaissäilytystä missä itse datan säilyttämisen lisäksi erityisen tärkeää on myös aineiston ymmärrettävyyden säilyttäminen. Näissä pitkäaikaissäilytyspalveluissa noudatetaan usein kansainvälisiä standardeja, jotka tekevät palveluista keskenään vertailukelpoisia ja edesauttavat palveluiden välisten integraatioiden kehittämistä.

Tämä luku esittelee pitkäaikaissäilytyksessä käytettyjä standardeja aloittaen korkealla tasolla pitkäaikaissäilytyspalvelun OAIS-viitemallin mukaisesta funktionaalisesta mallista ja siirtyen yksittäisiä aineistoja kuvaileviin METS- ja PREMIS-standardeihin keskittyen erityisesti siihen, miten standardit käsittelevät yksittäisiä tiedostoja ja niiden käyttämiä tiedostomuotoja. Lopuksi esitellään esitettyjä standardeja hyödyntäviä pitkäaikaissäilytyspalveluja ja -ohjelmistoja.

2.1 Standardit

Pitkäaikaispalvelujen suunnittelussa ja toteutuksessa käytetään standardeja, joilla pyritään tekemään palveluista keskenään yhdenmukaisempia ja vertailukelpoisia. Näitä standardeja noudattamalla voidaan suunnitella ja kehittää ohjelmistoja, jotka esimerkiksi mahdollistavat aineistojen siirtämisen

ja kopioinnin helpommin eri pitkäaikaissäilytyspalvelujen välillä.

Nämä standardit voivat koskea pitkäaikaissäilytyspalveluja korkealla tasolla (mikä on "säilytettävä aineisto"?), sekä käsitellä alhaisemmalla tasolla esimerkiksi yksittäisten tiedostojen teknisiä piirteitä ja metatietoja (miten ilmaistaan videotiedoston sisältämät tekniset piirteet?).

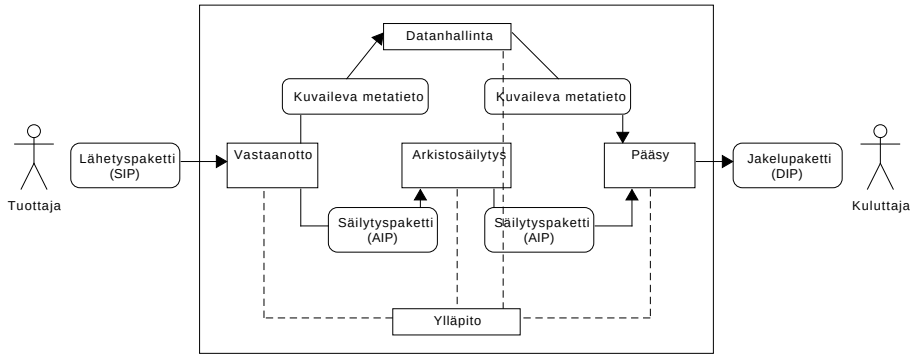
OAIS-viitemalli (Open Archival Information System) määrittelee pitkäaikaissäilytyspalvelun funktionaalisen mallin ja sen sisällä käytettävät termit[Lav00]. Sama viitemalli on kuvattu myös ISO 14721:2003 -standardissa. OAIS-mallissa säilytettävät aineistot ilmaistaan kolmea pakettityyppiä ja niiden elinkaarta käyttäen: siirtopaketti (SIP, Submission Information Package), säilytyspaketti (AIP, Archival Information Package) ja jakelupaketti (DIP, Dissemination Information Package).

Aineiston säilytysprosessi alkaa, kun aineiston tuottaja paketoi yksittäisen aineiston ja sitä koskevan metatiedon siirtopaketiksi. Tämä siirtopaketti lähetetään pitkäaikaissäilytyspalveluun, joka vastaanottovaiheessa tarkistaa siirtopaketin ja muuntaa sen pitkäaikaissäilytykseen sopivaan muotoon eli säilytyspaketiksi.

Jos aineisto läpäisee vastaanottovaiheen ja muuttamisen säilytyspaketiksi, pitkäaikaissäilytyspalvelu säilyttää paketin ja vastaa sen ylläpidosta, johon voi kuulua muun muassa aineiston replikointi fyysisille medioille, ajoittainen virheentarkistus ja tarvittaessa siirto uusiin tallennusmedioihin. Aineiston ollessa säilytyksessä kuluttaja voi hakea siitä kopion, mikä tapahtuu palvelussa pääsyvaiheen avulla. Tämän päätteeksi kuluttaja saa jakelupakettina kopion arkistoidusta aineistosta. OAIS-palvelusta on esitetty funktionaalinen malli kuvassa 1.

Esitelty OAIS-viitemalli ei määrittele tarkasti mitä kaikkea esimerkiksi säilytyspaketti voi sisältää, ja miten sen sisältämä aineisto ilmaistaan.

METS (Metadata Encoding and Transmission Standard) on standardi, joka määrittelee korkealla tasolla, miten aineistoon kuuluvat tiedostot ja niiden metatiedot ilmaistaan [MET]. METS käsittää aineiston sisältämien tiedostojen lisäksi myös muun muassa hallinnollisia tietoja kuten aineiston tekijänoikeudet ja suhteet toisiin aineistoihin. METS ei kuitenkaan määrittele tarkasti, mitä tiedostojen teknisiä piirteitä asiakirjassa pitää olla ja miten ne tulee ilmaista. Sen sijaan METS sallii muiden standardien mukaisen metatiedon sisällyttämisen samaan metatietoasiakirjaan. Standardia kehitetään osana *Digital Library Federation* -aloitetta. Esimerkki METS-asiakirjan



Kuva 1: Funktionaalinen malli OAIS-mallin mukaisen arkistointipalvelun eri toiminnoista[Lav00]. Kuvassa käytetään Suomen kansallisten pitkäaikaissäilytyspalvelujen laatimia suomennoksia[PASa].

otsikosta on esitetty listauksessa 1.

```
<metsHdr CREATEDATE="2003-07-04T15:00:00"
         RECORDSTATUS="Complete">
    <agent ROLE="CREATOR" TYPE="INDIVIDUAL">
        <name>Jerome McDonough</name>
    </agent>
    <agent ROLE="ARCHIVIST" TYPE="INDIVIDUAL">
        <name>Ann Butler</name>
    </agent>
</metsHdr>
```

Listaus 1: Esimerkki METS-asiakirjan otsikosta, jossa ilmenee METS-asiakirjaa koskevan aineiston luontipäivä ja aineiston säilytyksen kannalta oleelliset tahot (aineiston luoja sekä sen arkistoinut henkilö). METS-asiakirjat käyttävät XML-asiakirjaformaattia, mikä mahdollistaa METS-asiakirjojen koneellisen käsittelyn.

Tiedostoja tarkemmin kuvaaviin standardeihin kuuluu muun muassa *PREMIS* (Preservation Metadata: Implementation Strategies), joka määrittelee tarkoin, miten aineistoa koskevat metatiedot tallennetaan ja käsitellään. Kyseinen standardi keskittyy erityisesti “toteutettavissa olevaan metatietoon”, joka on tarkasti määriteltävissä ja toteutettavissa ja jonka luonti ja käsittely onnistuu automatisoidusti [C+08]. Täten PREMIS käsittelee

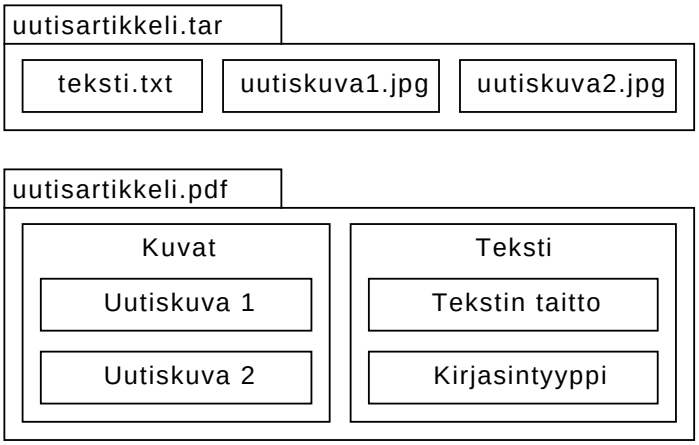
pääasiassa tiedostojen teknisiä piirteitä, jotka voidaan lukea tiedostosta automaattisesti tai muuten sisällyttää osana jotain automaattista työvuota. PREMIS-standardi syntyi *Online Computer Library Center* ja *Research Libraries Group* -organisaatioiden tukeman *PREMIS Working Group* -työryhmän tuloksena.

Sekä PREMIS- ja METS-asiakirjat käyttävät XML-asiakirjamuotoa, joka mahdollistaa metatietojen koneellisen käsittelyn ja validoinnin XML-skeeman avulla. Esimerkiksi METS- ja/tai PREMIS-asiakirjojen oikeamuotoisuus on mahdollista tarkistaa XML-validaattorilla, jolle on annettu syötteenä asiakirja ja XML-skeema. Pitkäaikaissäilytyspalvelulla saattaa olla myös omia määritelmiä ja vaatimuksia, joiden takia esimerkiksi jotkin PREMIS-elementit saattavat olla pakollisia tai kiellettyjä. Tässä tapauksessa palvelun tarvitsee jakaa ainoastaan muokattu XML-skeema, jonka avulla palvelukohtainen METS- tai PREMIS-validointi on mahdollista muokkaamatta itse XML-validaattoria. Tämä edesauttaa yhteensopivien pitkäaikaissäilytysohjelmistojen kehittämistä.

Edellä mainittuja METS- ja PREMIS-standardeja, sekä OAIS-viitemallia noudattaa Kansallisen digitaalisen kirjaston aloittama pitkäaikaissäilytyspalvelu [LHK+15]. Palvelu noudattaa OAIS-mallia, joka vastaa yleisluontoisesta sanastosta mutta ei teknisestä toteutuksesta. Esimerkiksi siirtopaketti on käytännössä tiedostojärjestelmään luotu hakemisto, säilytyspaketti TAR-muodossa oleva arkisto, joka sisältää siirtopakettihakemiston, ja jakelupaketti REST-rajapinnan kautta ladattavissa oleva kopio säilytyspaketista. OAIS-mallin vastaanotto- ja pääsyvaiheet on toteutettu käyttäen mikropalveluja hyödyntäävä työvuota, joka muodostaa siirtopaketista eri työvaiheiden jälkeen arkistoitavan säilytyspaketin. Lisäksi pitkäaikaissäilytyspalvelulla on omat vaatimuksensa kelvollisten tiedostomuotojen suhteen: esimerkiksi arkistotiedostomuotojen (esimerkiksi TAR) tai kopiosuojausmenetelmien käyttö on kielletty, vaikka PREMIS-standardi pystyy ilmaisemaan nämä piirteet: esimerkiksi käyttörajoitettujen tiedostojen kuvailu onnistuu metatiedoissa esto-ominaisuuksien avulla (inhibitor)[C+08].

2.2 Aineistot

METS-standardissa yksi säilytettävä aineisto voidaan luokitella digitaaliseksi kirjastokappaleeksi (digital library object)[MET]. Yksi kirjastokappale ei kuitenkaan välttämättä vastaa vain yhtä tiedostoa, vaan voi tarpeen tullen



Kuva 2: Yksittäinen aineisto on paketoitavissa monin eri tavoin. Esimerkiksi uutisartikkeli, johon kuuluu kaksi uutiskuvaa, on esitetty tässä tekstitiedoston ja kaksi kuvatiedostoa sisältävänä TAR-arkistona sekä PDF-asiakirjana, joka sisältää tekstin ja kuvien lisäksi myös kirjasintyypin ja tekstin taittoon eli esitysmuotoon liittyvää tietoa.

koostua useista tiedostoista tai tiedostoryhmistä. Esimerkiksi uutisartikkeli, joka sisältää tekstiosion ja muutaman uutiskuvan, on mahdollista tulkita yhtenä digitaalisena kirjastokappaleena, mutta mahdollista säilyttää useana tiedostona. *METS* käyttää tätä tilannetta varten käsitettä tiedostoryhmä, joka METS-asiakirjassa ilmaistaan `<fileGrp>` -elementtinä. Uutisartikkelin tapauksessa `<fileGrp>`-elementti voisi sisältää usean version kyseisestä kirjastokappaleesta: alkuperäisen DOCX-tiedostomuodossa tallennetun ja muokattavassa muodossa olevan alkuperäiskopion, PDF-tiedostomuodossa olevan julkaisumuotoisen asiakirjan, ja TAR-arkiston, joka sisältää uutisartikkelin tekstin ja kuvat erillisinä tiedostoina, joita on helpompi käsitellä koneellisesti. Tätä tilannetta on havainnollistettu kuvassa 2.

2.3 Tiedostomuodot

METS-standardi määrittelee kirjastokappaleen ja tiedoston, mutta ei ota kantaa tämän tarkemmin tiedoston rakenteeseen. PREMIS-standardi sen sijaan sisältää määritelmän tiedostoille (file), tiedoston sisäisille bittivir-

roille (bitstream), sekä tiedostovirroille, jotka ovat bittivirtoja, jotka voi sellaisenaan muuttaa luettaviksi tiedostoiksi (filestream) [C+08]. Esimerkiksi jos aiemmin esitelty TAR-arkisto sisältää TXT-päätteisen tekstitiedoston ja kaksi JPG-tiedostomuotoista kuvatiedostoa ja kyseinen arkisto ei käytä pakkausta, sen voi katsoa sisältävän kolme bittivirtaa, jotka ovat myös tiedostovirtoja. Jos TAR-arkiston sisältö on pakattu, sen sisältämästä bittivirrasta pitää purkaa ensin tiedostojen alkuperäiset tiedostovirrat, ennen kuin ne ovat käyttökelpoisia.

Tiedostomuoto on yleensä tunnistettavissa joko tiedostonimen päätteestä tai tiedoston bittivirran alussa sijaitsevasta taikanumerosta; esimerkiksi PDF-asiakirjat alkavat `%PDF` tekstinpätkällä, joka heksadesimaalimuodossa ilmaistuna on `25 50 44 46`[Kli17]. Tiedostomuodon tunnistaminen ei aina ole yksiselitteistä riippuen tiedostomuodosta. Esimerkiksi arkistointitarpeisiin suunniteltu *PDF/A* -versio PDF-asiakirjamuodosta sisältää teknisiä vaatimuksia, jotka PDF-asiakirjan tulee täyttää. Näihin vaatimuksiin kuuluvat muun muassa se, että PDF-tiedoston täytyy sisältää sen käyttämät kirjasintyypit ja että asiakirjan rakenteen tulee noudattaa asiakirjan loogista rakennetta niin tarkasti kuin mahdollista. Tässä tapauksessa asiakirjan *PDF/A*-muotoisuus ei ilmene pelkästään tiedoston otsikkoa tai päätettä lukemalla.

Pitkäaikaissäilytyspalvelut voivat määritellä itse, mitä tiedostomuotoja palvelut hyväksyvät säilytykseen. Esimerkiksi kansallisille pitkäaikaispalveluille on määritelty säilytys- ja siirtokelpoiset tiedostomuodot [PASb]. Näiden valintaperiaatteiksi kuuluu tiedostomuotojen käyttökelpoisuuden lisäksi myös niitä käyttävien aineistojen lukumäärät. Tässä tapauksessa siirtokelpoisten tiedostomuotojen arvioidaan soveltuvan huonommin pitkäaikaissäilytykseen, mutta koska kyseisiä tiedostomuotoja käyttävää aineistoa on runsaasti, ne kannattaa silti säilyttää ensin ja muuntaa vasta myöhemmin säilytyskelpoiseen tiedostomuotoon.

2.4 Ohjelmistot

Digitaalinen pitkäaikaissäilytys vaatii toteutuakseen ohjelmistoja, joilla säilytyksen eri työvaiheet voidaan toteuttaa. Näihin työvaiheisiin voi kuulua aineiston vastaanottaminen, tunnistaminen, validointi ja migraatio toiseen tiedostomuotoon. Pitkäaikaissäilytyksessä käytetyt ohjelmistot voivat vaihdella huomattavasti, ja ne voivat vaihdella laajassa käytössä olevista tie-

dostomuodon käsittelytyökaluista erityisesti säilytystarpeisiin kehitettyihin ohjelmistoihin. Näitä ohjelmistoja on kerätty *COPTR* -wikiin (*Community Owned Digital Preservation Tool Registry*), jossa ne on lajiteltu muun muassa toiminnallisuuden perusteella eri kategorioihin[COP]. Näistä kategorioista mielenkiintoisia tutkielman kannalta ovat muun muassa tiedostomuotojen migraatio (File Format Migration), validointi (Validation) ja metatiedon purkaminen tiedostoista (Metadata Extraction).

COPTR-wikistä ei kuitenkaan ilmene, kuinka laajassa käytössä ohjelmistot ovat. Käyttöasteen selvittäminen on hankalaa, mutta lähtökohtana ohjelmistojen käyttöasteelle voidaan hyödyntää iPRES-konferenssien (*International Conference on Digital Preservation*) artikkeleita, ja sitä, kuinka monta kertaa ja kuinka monena vuonna niissä on käsitelty tiettyä ohjelmistoa. Tätä varten aiemmin mainittujen kategorioiden piiriin kuuluvista ohjelmistoista kerättiin lista nimiä. Näille nimille laskettiin hakuosumat konferenssien artikkeleissa vuosina 2010-2018 ottaen huomioon sekä mainintojen kokonaismäärän että niiden hajonnan eri vuosina. Oleellista on, että ohjelmistoa käsitellään vakituisesti useissa iPRES-konferenssin artikkeleissa.

Pöytäkirjoista mainituista ohjelmistoista suosituimmaksi osoittautui *Archivematica*, joka mainittiin vuotta 2011 lukuunottamatta jokaisessa konferenssissa. *Archivematica* on vapaan lähdekoodin ohjelmistokokoelma, joka noudattaa OAIS-viitemallia ja tukee mainittujen METS- ja PREMIS-standardien lisäksi muun muassa *Dublin Core* ja *Bagit* -määritelmiä [Arc18]. [1]

Archivematica on kehitetty mikropalveluarkkitehtuuria noudattaen. *Archivematica* hoitaa tiedostomuotojen erittelyn *Format Policy Registry* (FPR) -rekisterin avulla, joka määrittelee tiedostomuotojen kuvaukset, versiot ja soveltuuko tiedostomuoto loppukäyttäjien käytettäväksi ja/tai pitkäaikaissäilytykseen.

Muita usein mainittuja ohjelmistoja ovat muun muassa tiedostomuotojen tunnistamista varten kehitetty *DROID* [DROa], säilytettävän aineiston löytämiseen ja analysointiin soveltuva *BitCurator* [Bit], sekä tiedostomuotojen validointia varten kehitetyt *JHOVE* [JHOb] ja *JHOVE2* -ohjelmistot [JHOe].

JHOVE- ja JHOVE2-ohjelmistojen kohdalla huomionarvoista on ohjelmistojen kehityshistoria. JHOVE-ohjelmiston viimeisin versio on julkaistu vuonna 2018, ja kyseisen ohjelmiston kehitys on jatkunut *Open Preservation*

[1] Archivematica-ohjelmistokokoelman lähdekoodi on saatavilla GitHub-palvelussa: `https://github.com/artefactual`

Foundation -järjestön alaisuudessa. Kyseinen ohjelmisto on myös iPRES-mainintojen perusteella edelleen käytössä.

JHOVE2-ohjelmisto oli suunniteltu kokonaan uudeksi ohjelmistoksi, jonka tarkoituksena oli korjata alkuperäisessa JHOVE-ohjelmistossa olevia puutteita muun muassa eriyttämällä tiedoston tunnistaminen ja validointi [JHOe]. Ohjelmiston kehitystä rahoitettiin vuoteen 2011 asti, ja ohjelmiston viimeisin versio on julkaistu vuonna 2013, mutta lähdekoodivarastossa ei ole esiintynyt muutoksia kyseisen vuoden jälkeen [JHOd]. Myös iPRES-konferenssien artikkeleissa ei ole mainittu kyseistä ohjelmistoa vuoden 2013 jälkeen, kun taas sitä edeltävinä vuosina 2010-2012 ohjelmisto on mainittu jokaisena vuonna.

Kaikkien mainittujen ohjelmistojen lähdekoodi on vapaasti saatavilla, mikä mahdollistaa ohjelmistojen kehityshistorian analysoimisen.

3 Säilytyksen ongelmatilanteet

Digitaalinen pitkäaikaissäilytys vaatii aineiston koko elinkaaren aikana erilaisia toimenpiteitä. Kun aineisto on taltioitu ja siirretty pitkäaikaissäilytykseen, aineiston säilyminen ja ymmärrettävyys pitää myös säilyttää koko ajanjakson ajan; tämä voi edellyttää esimerkiksi tiedostojen ja metatiedon kopioimista vanhoista tallennusmedioista uusille tallennusmedioille, sekä aineiston muuttamisen uusiin tiedostomuotoihin.

Tässä luvussa esitellään tiedostomuodoissa mahdollisesti esiintyviä ongelmatilanteita, ja missä yhteydessä ne voivat vaarantaa aineiston säilyvyyden.

3.1 Puuttuva metatieto

Säilytetystä aineistosta voi menettää ymmärrettävyyden, jos sitä koskevaa metatietoa puuttuu. Esimerkiksi paljas teksti ja sen kattava bittivirta voidaan tallentaa sellaisenaan, mutta ilman tietoa tekstin merkistöstä sen esittäminen jälkeenpäin voi epäonnistua. Tätä tilannetta on havainnollistettu esimerkillä kuvassa 3.

Alkuperäinen teksti UTF-8 -merkistöä käyttäen:

```
Albert osti fagotin ja töräytti puhkuvan melodian.
```

UTF-8 -teksti esitetty ASCII-merkistöä käyttäen:

```
Albert osti fagotin ja t��r��ytti puhkuvan melodian.
```

UTF-8 -teksti esitetty ISO 8859-2 -merkistöä käyttäen:

```
Albert osti fagotin ja tĂśrĂ¤ytti puhkuvan melodian.
```

Kuva 3: Kuvan ensimmäinen lause on tallennettu UTF-8 -merkistöä käyttävänä bittivirtana. Bittivirta itsessään ei kuitenkaan riitä, ellei käytettyä merkistöä ole sisällytetty tiedoston rakenteeseen tai erillisiin metatietoihin. Jos säilytettyä tekstiä yritetään esittää väärää merkistöä käyttäen, teksti voi sisältää väärin tulkittuja tai puuttuvia merkkejä, jotka haittaavat loppukäyttäjän kykyä käyttää aineistoa.

Käytännössä esimerkin virhetilanne on helposti estettävissä. Puutteellisen aineiston säilyttäminen on estettävissä vaatimalla tekstitiedoston lisäksi tekstitiedoston merkistö erillisenä metatietona. Esimerkiksi kansallisten pitkäaikaissäilytyspalvelujen määrittelyssä paljaan tekstitiedoston merkistö pitää

ilmaista PREMIS-metatiedoissa mediatyyppiä (Internet media type/MIME) käyttäen[PASb]. Esimerkin tapauksessa kelvollinen arvo olisi `text/plain; charset=UTF-8`. Pitkäaikaissäilytysjärjestelmä varmistaisi tämän merkistötiedon olemassaolon siirtopaketin vastaanottovaiheessa; jos tietoa ei löydy, paketti hylätään, ja aineiston tuottaja saa ilmoituksen, josta ilmenee kyseisen metatiedon puuttuminen.

3.2 Tiedostomuodon päivitys

Pitkäaikaissäilytyksessä olevat tiedostomuodot pyritään valitsemaan aineiston pitkä elinkaari huomioon ottaen. Tästä huolimatta on mahdollista, että aineisto joudutaan muuntamaan eri tiedostomuotoon esimerkiksi, kun vanhasta tiedostomuodosta on tullut vaikeasti luettava.

Tämä toimenpide tuli tarpeelliseksi Uuden-Seelannin kansalliselle kirjastolle [GM14]. Kirjastolla oli hallussa WordStar-tiedostomuotoja käyttäviä asiakirjoja, joiden käyttö sellaisenaan vaati vanhentuneella *MS DOS* -käyttöjärjestelmällä toimivan tekstinmuokkausohjelmiston. Kirjasto kehitti tiedostomuodon migraatiota varten työkalun, joka muuttaa WordStar-muotoiset asiakirjat HTML 4 -tiedostomuotoon.

Migraatiotyökalua kirjoitettaessa tärkeäksi kysymykseksi muodostui, mitä alkuperäisten asiakirjojen piirteitä pyrittiin säilyttämään. WordStar-asiakirjat on suunniteltu muokattavaksi saman nimisellä ohjelmistolla, mutta loppukäyttäjät ovat kiinnostuneita vain asiakirjan tulostetusta versiosta, joka eroaa esitysmuodoltaan tekstinmuokkausohjelmistossa näkyvästä versiosta. Koska alkuperäiset asiakirjat oli tarkoitettu käytettäviksi valmiina tulosteina, migraatiotyökalun kehityksessä päädyttiin tulostetun asiakirjan tyylittelyn jäljittelemiseen. Toisaalta tässä muunnoksessa menetetään myös piirteitä, jotka olisivat hyödyllisiä käsiteltäessä tiedostoa tekstinmuokkausohjelmalla. Alkuperäisistä asiakirjoista löydettiin merkkejä, jotka ilmaisevat tulostimelle lähetettäviä koodeja. Koska näiden koodien merkitystä ei pystytty selvittämään, eivätkä ne vaikuttaneet lopulliseen tulostettuun kopioon, näitä koodeja ei käsitellä muunnoksessa.

3.3 Tiedostorajoitukset

Pitkäaikaissäilytykseen lähetetty aineisto saattaa sisältää rajoituksia, jotka rajoittavat tai estävät aineiston käytön [C$^+$08]. Näihin kuuluvat muun muassa

salasanasuojaus ja erilaiset digitaaliset käyttöoikeuksien hallintamenetelmät (DRM).

Saksan kansallinen kirjasto kehitti DRM-menetelmien tunnistamista varten työkalun, joka tunnistaa ja arvioi aineiston sisältämät kopionsuojausmenetelmät [HS14]. Tämän arvion perusteella aineisto sijoitetaan johonkin pitkäaikaissäilytyksen vaaraluokkaan (*Long-Term-Preservation Risk*, LTPR). Yksinkertaisimmat kopionsuojausmenetelmät lisäävät aineistoon piilotettuja "vesileimoja", jotka eivät estä aineistojen käyttämistä tai levittämistä, mutta mahdollistavat aineiston laittoman levittämisen seuraamisen. Näitä kopionsuojausmenetelmiä käyttävät aineistot voidaan hyväksyä pitkäaikaissäilytykseen, sillä ne eivät vaaranna pitkäaikaissäilytyksen toteutumista.

Korkeamman vaaran sisältävät kopionsuojausmenetelmät voivat sen sijaan olla riippuvaisia ohjelmistokomponenteista tai järjestelmistä, kuten ulkoisista palvelimista, jotka tarkastavat tiedostojen käyttöoikeudet ennen niiden esittämistä käyttäjälle. Tällaisten tiedostojen hyväksyminen voi vaarantaa tiedostojen ymmärrettävyyden, jos esimerkiksi kopiosuojauksesta vastaavat ohjelmistokomponentit tai ulkoiset palvelut lakkaavat toimimasta.

Saksan kansallinen kirjasto käyttää työkalua kopionsuojausmenetelmien tunnistamiseen vastaanottovaiheessa. Koska pitkäaikaissäilytysjärjestelmät toimivat pitkälti automaattisesti, työkalu estää tilanteet, jossa pitkäaikaissäilytykseen lähetetään materiaalia, jonka säilyvyyttä ei todellisuudessa voi taata. Tässä tapauksessa on tärkeätä tunnistaa aineistossa olemassa olevat kopiosuojausmenetelmät mahdollisimman aikaisin, jotta aineiston toimittajalta voidaan pyytää pitkäaikaissäilytykseen soveltuva versio aineistosta, joka ei sisällä kopiosuojausmenetelmiä. Mitä pidempi aikaväli on kulunut aineiston ottamisesta säilytykseen, sitä vaikeampaa on saada yhteys aineiston oikeudenhaltijaan.

3.4 Ohjelmistopatentit

Ohjelmistopatentit voivat rajoittaa erilaisten tiedostomuotojen tai niiden käyttämien menetelmien saatavuutta. Patentit voivat olla maakohtaisia, Esimerkiksi TIFF-tiedostomuodossa oli alunperin LZM-pohjainen (*Lempel-Ziv-Welch*) pakkausmenetelmä, joka kuitenkin poistettiin Baseline TIFF -määritelmästä, kun ilmeni, että pakkausmenetelmää koskee ainakin yksi rekisteröity patentti [TIF]. Tämä voi hankaloittaa tiedostomuotojen käsittelyä, jos patentin alaista tiedostomuotoa ei voi käyttää järjestelmässä; esimer-

kiksi Linux-pohjainen käyttöjärjestelmä Ubuntu ei oletuksena sisällä tukea erilaisiin “ei-vapaisiin” formaatteihin patentti ja tekijänoikeussyistä [Ubu]. *Alliance for Open Media* -järjestön kehittämä *AV1* on tekijänoikeuskorvausvapaa videokoodekki, joka on suunniteltu siten, ettei se ole ristiriidassa olemassaolevien ohjelmistopatenttien kanssa.

4 Tiedostomuodot

Pitkäaikaissäilytyspalvelut määrittelevät palveluun hyväksyttävät tiedostomuodot. Esimerkiksi Suomen kansallinen pitkäaikaissäilytyspalvelu määrittelee pitkäaikaissäilytykseen hyväksyttävät tiedostomuodot ja kyseisten tiedostomuotojen jaottelun joko säilytys- tai siirtokelpoisiin tiedostomuotoihin [PASb]. Nämä määritelmät voivat sisältää useita tiedostomuotoja, jotka käytännössä täyttävät samat vaatimukset: esimerkiksi sekä PDF (Portable Document Format) ja HTML (Hypertext Markup Language) -tiedostomuodot mahdollistavat kuvien ja muiden piirteiden tallentamisen itse tekstisisällön lisäksi. Tiedostomuotojen lajittelu eri kategorioihin saattaa vaihdella pitkäaikaissäilytyspalvelusta riippuen. Kansallinen pitkäaikaisäilytyspalvelu lajittelee sekä HTML että PDF -tiedostomuodot *Teksti*-kategorian alle[PASb]. Archivematica sen sijaan sisältää tarkemmat lajittelut eri tekstitiedostomuodoille: tekstitiedosto voi tiedostomuodosta riippuen olla strukturoitua (*Text (Structured)*, olla lähdekoodia *Text (Source Code)* tai paljasmuotoista tekstiä *Text (Plain)*.

Tässä luvussa käsitellyt tiedostomuodot ovat kansallisen pitkäaikaissäilytyspalvelun käyttämiä, ellei toisin ole mainittu.

4.1 Asiakirjat

Asiakirjalla tarkoitetaan tässä yhteydessä tekstiä sekä staattisia elementtejä, kuten kuvia, kirjasintyyppejä ja värejä. Kyseessä voi olla esimerkiksi kopio tieteellisestä artikkelista, joka voi olla muokattavissa sellaisenaan tai olla jo valmiissa julkaisumuodossa.

PDF (*Portable Document Format*) on laajassa käytössä oleva asiakirjamuoto, jonka tavoitteena on pystyä esittämään asiakirja samassa muodossa riippumatta laitteesta tai esitystavasta: tämä voi tarkoittaa esimerkiksi tulostettua kopiota tai mobiililaitteella avattua kopiota asiakirjasta [Kli17]. Käytännössä sama asiakirja voi poiketa visuaalisesti tekijän alkuperäiskappaleesta esimerkiksi, jos laitteesta puuttuu asiakirjan käyttämä kirjasintyyppi, joka ei sisälly PDF-tiedostoon.

PDF-tiedostomuotoa ja *Adobe Reader* -ohjelmistoa kehittänyt *Adobe* on vuosien kulussa laajentanut PDF-standardia vapaaehtoisilla ominaisuuksilla, kuten web-linkeillä, salauksella ja video-objekteilla. Nämä ominaisuudet mahdollistavat PDF-asiakirjojen käytön monissa uusissa käyttötapauksissa,

mutta toisaalta vaikeuttavat kyseisten asiakirjojen säilyttämistä: esimerkiksi videotiedoston sisällyttäminen vaatii, että asiakirjan tekstisisällön esittämisen lisäksi ohjelma osaa toistaa myös asiakirjan sisältämät videotiedostot.

PDF/A on pitkäaikaiseen arkistointiin suunniteltu versiokokoelma PDF-asiakirjamuodosta, joka määrittelee tekniset vaatimukset, jotka PDF-asiakirjan tulee täyttää. Näillä vaatimuksilla pyritään säilyttämään PDF-asiakirjan toistettavuus ottaen huomioon mahdolliset muutokset järjestelmässä; esimerkiksi käyttöjärjestelmän käyttämillä kirjasintyypeillä ei tulisi olla vaikutusta asiakirjan ulkoasuun, sillä tarpeelliset kirjasintyypit sisältyvät PDF/A-muotoiseen asiakirjaan.

PDF/A ei viittaa yhteen ainoaan määritelmään, vaan vuosien varrella on julkaistu useita versioita, joilla on omat vaatimuksensa. Esimerkiksi vuonna 2005 ISO-standardoitu [ISO05] (ISO/IEC 19005-1:2005) PDF/A perustuu PDF 1.4 -versioon ja sisältää kaksi vaatimustasoa: PDF/A-1a vaatii muun muassa asiakirjan käyttämien kirjasintyyppien sisällyttämisen PDF-tiedostoon, hierarkkisen asiakirjarakenteen sekä kuvaavan tekstin jokaiselle asiakirjassa esiintyvälle kuvalle. Saman standardin alle kuuluva PDF/A-1b vaatii ainoastaan kirjasintyyppien sisällyttämisen asiakirjaan. PDF/A-2 ja PDF/A-3 -versiot on julkaistu vuosina 2011 ja 2012 ja perustuvat uudempiin PDF-versioihin, eivätkä ne ole takaisinpäin yhteensopivia.

HTML (*Hypertext Markup Language*) on www-sivujen tuottamiseen ja esittämiseen luotu merkintäkieli, jonka syntaksi muistuttaa paljon XML-asiakirjoissa käytettyä syntaksia. HTML-kielestä on olemassa monia versioita: vuonna 1999 julkaistu HTML 4.01 on julkaistu myös ISO-standardina [ISO00] (ISO/IEC 15445:2000), sekä vuonna 2014 julkaistu HTML 5 [PASb]. HTML-kielellä on mahdollista ilmaista sivun semanttista rakennetta: HTML 4 -kielen otsikkoelementtien `<h1>, <h2>, ..., <h5>` ja asiakirjan osat erottavien elementtien `<div>` avulla on mahdollista luoda asiakirja, josta on mahdollista johtaa sisällysluettelo [HTMb]. HTML 5 -kieleen kuuluvat elementit kuten `<article>` ja `<section>` mahdollistavat tekstin rakenteen ilmaisemisen vielä tarkemmin: kyseisillä elementeillä on mahdollista ilmaista myös HTML-sivun sisältämät artikkelit ja itsenäiset tekstiosuudet.

Nämä ominaisuudet auttavat asiakirjan tulkintaa, mutta siitä saatavat hyödyt eivät rajoitu pelkästään metatiedon keräämiseen säilytystä varten. Esimerkiksi esteettömysohjelmistot voivat auttaa näkövammaisia käyttämään web-sivuja, jos sivu sisältää esteetöntä käyttöä varten vaadittavat elementit.

HTML-asiakirja ei itsessään sisällä resursseja kuten sivun tyylittelyjä ja kuvia. Sen sijaan HTML voi sisältää viittauksia ulkoisiin resursseihin, jotka web-selain voi ladata rinnakkain. Näihin ulkoisiin resursseihin voi kuulua muun muassa kuvia, tyylittelyjä (käyttäen CSS-kieltä (*Cascading Style Sheets*) ja skriptitiedostoja (käyttäen JavaScript-kieltä), joilla on mahdollista lisätä web-sivuun interaktiivisia elementtejä. Pitkäaikaissäilytystä varten nämä resurssit täytyy pakata yhtenäiseksi aineistoksi: tämä on mahdollista esimerkiksi lisäämällä kaikki web-sivua koskevat tiedostot yhteen TAR-arkistoon.

ODF (*Open Document Format*) on XML-pohjainen tiedostomuoto, joka on suunniteltu tekstin käsittelyyn sekä muun muassa taulukkolaskentaan, esityksiin; näistä käyttötapauksista käsitellään pääasiassa tekstin käsittelyä [OAS]. Tiedostomuodon pääte riippuu ODF-asiakirjan käyttötarkoituksesta: esimerkiksi tekstiasiakirjat käyttävät `.odt` -päätettä ja taulukkoasiakirjat `.ods` -päätettä. HTML-tiedostomuodosta poiketen ODF ei ole ihmisluettava sellaisenaan, vaan se käyttää ZIP-arkistopakkausta: tämän arkiston sisälle tallennetaan erillisinä tiedostoina asiakirjan sisältö, tyylittely ja muut resurssit. OASIS on julkaissut ODF-tiedostomuotostandardin version 1.2 vuonna 2011, joka on myös ISO-standardoitu vuonna 2015 [ISO15] (ISO/IEC 26300-1:2015) osana ISOn vapaasti saatavilla olevia standardeja [OAS].

PDF-asiakirjojen tavoin ODF sisältää asiakirjassa käytetyt resurssit, joten ODF-tiedosto soveltuu muokkaustarpeisiin. Kirjasinlajit sen sijaan eivät sisälly ODF-asiakirjaan, mutta LibreOffice-ohjelmisto sallii tekstiasiakirjojen tallentamisen aiemmin mainittuun PDF/A-1a -muotoon. Tässä tapauksessa ODF-tiedostomuoto soveltuu käytettäväksi silloin, kun asiakirjaa halutaan muokata tai säilyttää asiakirjassa olevia piirteitä, jotka eivät säily "julkaisumuotoisessa" PDF-asiakirjassa. Näihin piirteisiin voi kuulua muun muassa tekstisisällössä tehdyt lisäykset ja poistot, jotka ilmaistaan standardissa `<text:insertion>` ja `<text:deletion>` -elementeillä. PDF-asiakirjassa sen sijaan muokkaushistoria ei säilyisi.

4.2 Kuvat

Kuvilla tarkoitetaan tässä kategoriassa kaksiulotteisia värikuvia, jotka voivat olla valokuvia tai tietokoneella luotuja piirroksia. Kuvat tallennetaan digitaalisesti lukujonoina käyttäen jotakin värimallia: esimerkiksi RGB-mallissa jokaista kuvan pikseliä kohden on olemassa punaisen, vihreän ja sinisen värin

sävyä kuvaava numeerinen arvo.

Tämä värimalli pelkästään ei kuitenkaan ota kantaa laitteen tai fyysisen kopion kykyyn jäljitellä alkuperäisen kuvan väriä [ICC]. Tietokonenäyttö, jolla kuvaa on muokattu, saattaa näyttää kuvan värit hyvin eri tavalla kuin toinen tietokonenäyttö, jolla loppukäyttäjä katsoo tallennettua kuvaa. Tätä varten kuvatiedostoihin voi olla myös mahdollisuus tallentaa erillinen väriavaruus ICC-profiilia (*International Color Consortium*) käyttäen. Tämä ICC-profiili mahdollistaa mahdollistaa kuvan värien tarkan esittämisen riippumatta kohdelaitteen eroavaisuuksista värintoiston suhteen. Ellei laitteiden toisistaan eroavia värintoistokykyjä oteta huomioon, kuvan sisältämät värisävyt saattavat muuttua, kun sitä siirretään muodosta toiseen: esimerkiksi digitaalikamerasta kuvankäsittelyohjelmaan ja kuvankäsittelyohjelmasta tulostettuun kopioon.

Toinen tärkeä tekijä kuvan säilyttämisen kannalta on kuvatiedoston hyödyntämä pakkausmenetelmä. Kuvan pakkaus voi menetelmästä rippuen olla häviöllinen tai häviötön. Häviölliset pakkausmenetelmät yleensä vähentävät tiedostokokoa huomattavasti, mutta poistavat kuvasta pysyvästi informaatiota pyrkien kuitenkin säilyttämään näennäisesti samanlaatuisen kuvan. Jos häviöllisessä muodossa tallennettua kuvaa kuitenkin muokataan ja tallennetaan yhä uudelleen häviölliseen kuvatiedostomuotoon, kuvan informaatiota häviää jokaisella käsittelykerralla, kunnes kuvan laadun putoamisen voi havaita selkeästi. Tätä on havainnollistettu kuvassa 4. Häviöttömät pakkausmenetelmät pyrkivät vähentämään tiedostokokoa poistamatta alkuperäisessä kuvassa olevia yksityiskohtia; nämä menetelmät eivät vähennä kuvan tiedostokokoa yhtä rajusti. Tämä voi tehdä häviöttömistä pakkausmenetelmästä paremmin pitkäaikaissäilytykseen soveltuvan, jos häviöttömiä kuvia varten voidaan varata tarpeeksi tallennustilaa. Muussa tapauksessa myös häviöllisessä muodossa tallennettu kuva voidaan kelpuuttaa pitkäaikaissäilytykseen, jos sitä ei käsitellä uudelleen. Esimerkiksi *British Library* analysoi vuonna 2013 häviöttömästi pakattujen TIFF-kuvatiedostojen muuntamista häviöllistä pakkausta hyödyntävään JPEG2000-muotoon [PMC13]. Tässä tapauksessa migraatiosta saatavien hyötyjen voidaan katsoa olevan tärkeämpiä kuin tiedostomuodon muunnosprosessissa tapahtuva kuvan laadun heikkeneminen: kuvien tiedostokoko on pienempi ja saman tiedoston sisällä on mahdollista säilyttää usea kopio kuvasta eri käyttötarkoituksia varten.

JPEG (*Joint Photographic Experts Group*) tarkoittaa häviöllistä kuvan-

Kuva 4: Kun kuva tallennetaan häviöllistä pakkausta hyödyntävään tiedostomuotoon, kuvan sisältämää informaatiota katoaa, ja kuvan entropia kasvaa. Jos näin luotuja uusia kuvatiedostoja muokataan edelleen, kuvasta katoaa yhä enemmän informaatiota jokaisella tallennuskerralla. Tätä on simuloitu tässä kuvasarjassa, jossa kuvasarjan ensimmäistä kuvaa käännetään jokaisessa iteraatiossa 90 astetta ja näin luotu kuva tallennetaan häviöllisessä JPEG-muodossa uuteen tiedostoon. Kun tätä operaatiota on toistettu tarpeeksi monta kertaa, kuva muuttuu lopulta käyttökelvottomaksi; kuvasarjan viimeistä kuvaa on käsitelty 1280 kertaa, minkä aikana lähes kaikki kuvan sisältämä informaatio on kadonnut. Kuvasarjassa käsitelty alkuperäiskuva on julkaistu *Creative Commons Attribution-Share Alike 4.0 International* -lisenssin alaisena Wikimedia-sivustolla: `https://commons.wikimedia.org/wiki/File:Yellow-billed_shrike_(Corvinella_corvina_corvina).jpg`

pakkausmenetelmää sekä kuvatiedostomuotoa, joka käyttää kyseistä menetelmää [Wal92]. JPEG-muotoiset kuvatiedostot käyttävät `.jpeg` -tiedostopäätettä. Kyseinen tiedostomuoto on suosittu erityisesti valokuvien tallentamisessa ja siirtämisessä, ja suuri osa käytössä olevista laitteista ja ohjelmistoista, kuten digitaalikamerat ja web-selaimet, tukevat kyseistä tiedostomuotoa. JPEG-standardi julkaistiin vuonna 1992 ja tämän jälkeen ISO-standardina ISO 10918-1 vuonna 1994 [ISO94].

JPEG 2000 on JPEG-tiedostomuodon tavoin häviöllistä kuvanpakkausmenetelmää hyödyntävä kuvatiedostomuoto, joka kehitettiin JPEG-tiedostomuodon korvaajaksi [SCE01]. Standardin mukaiset kuvatiedostot käyttävät pää-

tettä `.jp2`. Standardi julkaistiin nimensä mukaisesti vuonna 2000 ja myöhemmin ISO-standardina 10918-1 [ISO94]. Alkuperäisestä JPEG-standardista poiketen JPEG 2000 ei ole kuitenkaan vakiintunut yleisessä käytössä: muun muassa suurin osa web-selaimista ei tue kyseistä kuvatiedostomuotoa [JPEa]. Tiedostomuodon käsittelyä varten on olemassa vapaan lähdekoodin *OpenJPEG* -kirjasto, mutta loppukäyttäjäsovelluksissa kuten kuvankatselu- ja kuvankäsittelysovelluksissa on muita käsiteltyjä kuvatiedostomuotoja huonompi tuki. Illinoisin yliopiston tekemässä tutkimuksessa merkittävä osa yliopiston säilömistä JPEG 2000 -kuvatiedostoista ei ollut luettavissa laajassa käytössä olevilla asiakasohjelmistoilla kuten *Adobe Photoshop* -kuvanmuokkaussovelluksella, joka tukee JPEG 2000 -tiedostomuotoa, mutta ei pystynyt avaamaan 57% testatuista JPEG 2000 -tiedostoista [RW16]. Huomioitavaa on, että kyseiset tiedostot oli avattavissa muilla ohjelmistoilla kuten avoimen lähdekoodin *ImageMagick* ja kaupallinen *Kakadu*. Yliopiston käyttämä FITS (*File Information Tool Set*) -ohjelmisto[2] myös tulkitsi kaikki tiedostot standardin mukaisiksi. Yliopiston tutkimustyön aikana ei ilmennyt, minkä takia osa tiedostoista ei toiminut kattavasti kaikilla testatuilla ohjelmistoilla, mutta tutkimus mainitsee, että standardin määritelmässä löytyy epäselviä kohtia, jotka ovat johtaneet tiedostomuotoa käyttävissä ohjelmistoissa ohjelmistokohtaisiin eroihin [VdK11].

TIFF (*Tagged Image File Format*) on kuvatiedostomuoto. Tiedostomuodon kehitti alunperin Aldus, joka on nykyisin osa Adobea [TIF]. Tiedostomuodon viimeisin versio 6.0 on julkaistu vuonna 1992, ja tämän määritelmän pohjalta on luotu myös muita TIFF-pohjaisia tiedostomuotoja: näihin kuuluvat muun muassa TIFF/EP (*Tagged Image File Format / Electronic Photography*) ja TIFF/IT (*Tagged Image File Format / Image Technology*), jotka ovat myös saatavilla ISO-standardeina (TIFF/EP: ISO 12234-2:2001 [ISO01] ja TIFF/IT: ISO 12639:2004 [ISO04a]).

Aiemmista JPEG-tiedostomuodoista poiketen TIFF ei määrittele ainoastaan yhtä kuvanpakkausmenetelmää, vaan TIFF on laajennettavissa muiden piirteiden lisäksi myös eri pakkausmenetelmillä. *Baseline TIFF* -määritelmä, johon kuuluu TIFF-standardin oleellisimmat piirteet, joita ohjelmistojen tulee tukea, määrittelee kolme pakkausmenetelmää: pakkaamaton data, *CCITT Group 3 1-Dimensional Modified Huffman* -menetelmä ja *PackBits* -menetelmä. JPEG-tiedostomuodosta poiketen nämä pakkaus-

[2]FITS tarkistaa tiedostot JHOVE- ja DROID-ohjelmistoja käyttäen.

menetelmät ovat häviöttömiä. TIFF-tiedostomuodolle on myös olemassa laajennus, joka sallii JPEG-kuvapakkauksen käytön TIFF-tiedostossa. Tämä laajennus on kuitenkin vapaaehtoinen eikä siten yhtä laajassa käytössä kuin aiemmin mainitut pakkausmenetelmät. Huomionarvoista on myös se, että Baseline TIFF -määritelmään kuului aikaisemmin myös LZM-pohjainen (*Lempel-Ziv-Welch*) häviötön pakkausmenetelmä. Tämä kuitenkin siirrettiin TIFF-laajennuksiin, kun ilmeni, että ainakin yksi Yhdysvalloissa rekisteröity patentti koskee kyseistä pakkausmenetelmää. Tämä Unisys-yrityksen omistama patentti on sittemmin rauennut vuonna 2003.

4.3 Ääni

Digitaalinen ääni tarkoittaa äänisignaalia, joka on muunnettu digitaaliseen muotoon numeerisina näytteinä käyttäen tiettyä taajuutta. Äänitiedosto voi koostua yhdestä tai useammasta äänisignaalista, jotka on tallennettu tiettyä näytetaajuutta ja bittinopeutta käyttäen. Esimerkiksi 16-bittinen 44,1 kHz -taajuudella tallennettu stereoäänitiedosto koostuu kahdesta äänisignaalista, jossa yhden sekunnin pituinen osuus ääniraidasta koostuu noin 44 tuhannesta ääninäytteestä, jossa jokaisen näytteen koko on 16 bittiä. Itse äänisignaalien lisäksi äänitiedosto voi sisältää muitakin tietoja: esimerkiksi musiikkikappaleen tapauksessa artistin nimen.

Kuvatiedostojen tavoin äänitiedostot ovat pakattavissa häviötöntä tai häviöllistä pakkausmenetelmää käyttäen. Äänitiedostoissa käytetyt pakkausmenetelmät kuitenkin toimivat eri periaatteita käyttäen, mikä mahdollistaa paremman pakkaussuhteen kuin yleiskäyttöisillä pakkausalgoritmeilla. Esimerkiksi kaksikanavaisissa äänitiedostoissa kanavat saattavat sisältää keskenään samanlaista ääni-informaatiota. Kanavien välistä korrelaatiota voidaan siten hyödyntää paremman pakkaussuhteen aikaansaamiseksi [FLAc].

FLAC (*Free Lossless Audio Codec*) on häviötöntä pakkausmenetelmää hyödyntävä audiotiedostomuoto sekä äänipakkausmenetelmä [FLAb]. FLAC-tiedostomuoto sisältää pakkausmenetelmän lisäksi myös muita säilytykseen soveltuvia piirteitä: FLAC-tiedostoon voi sisällyttää metatietoa koskien esimerkiksi aineiston tekijöitä sekä CD-kohtaisia metatietoja, jos äänitiedosto on kopioitu CD-levyltä. FLAC-tiedostot on tunnistettavissa `.flac` -päätteestä.

WAVE (*Waveform Audio File Format*) on Microsoftin ja IBM:n kehittämä audiotiedostomuoto [WAV]. WAVE perustuu RIFF-tiedostomuotoon (*Resource Interchange File Format*), joka määrittelee multimedia-informaation

säilöntään soveltuvan tiedostorakenteen. WAVE ja muut RIFF-tiedostorakennetta käyttävät tiedostomuodot jakavat datan useisiin lohkoihin. WAVE-tiedostomuoto tukee pakkausta, mutta useimmat käytössä olevat audiotiedostot ovat pakkaamattomia, minkä johdosta tiedostomuoto soveltuu käytettäviksi äänitiedostoja muokattaessa, missä tiedostovirran jatkuva purkaminen ja pakkaaminen vaikuttaisi ohjelman käytettävyyteen. Tiedostomuodossa on kuitenkin neljän gigatavun tiedostokokorajoitus, joka voi rajoittaa WAVE:n ja muiden RIFF-pohjaisten tiedostomuotojen käyttötapauksia. [3] Tällaisia saattavat olla esimerkiksi pitkäkestoiset ja pakkaamattomat äänitallenteet.

MP3 (*MPEG-1 Audio Layer III* tai *MPEG-2 Audio Layer III*) on vuonna 1993 ISO-standardoitu [ISO93] (ISO/IEC 11172-3:1993) häviöllistä pakkausmenetelmää hyödyntävän audiotiedostomuoto. Tiedostomuodon käyttämä äänenpakkausmenetelmä hyödyntää psykoakustisia malleja, joiden avulla äänivirtaa voidaan pakata tehokkaasti heikentämättä kuitenkaan ihmisen havaitsemaa äänen laatua. MP3-tiedostomuoto on laajassa käytössä useimmissa äänentoistoa tukevissa laitteissa ja ohjelmistoissa kuten web-selaimissa [MP3].

MP3 oli 23.5.2017 asti Technicolor-lisenssiohjelman alainen, mikä tarkoitti, että tiedostomuotoa koskevat toteutukset edellyttivät lisenssimaksuja. Tiedostomuotoa koskevat patentit ovat sittemmin rauenneet, mikä on sallinut tiedostomuodon käyttöönoton. Esimerkiksi Linux-pohjainen Fedora-käyttöjärjestelmä ei ennen patenttien raukeamista sisältänyt MP3-tukea [Fed]. Ennen tätä ajankohtaa MP3-tuen käyttöönotto vaati kolmannen osapuolen pakettien asentamista.

[3] RIFF-ylätunniste ilmaisee tiedoston koon käyttäen 32-bittistä etumerkitöntä kokonaislukua, mikä rajoittaa tiedoston enimmäiskoon 4,294,967,296 tavuun eli noin neljään gigatavuun.

5 Ohjelmistot tiedostomuotojen käsittelyssä

Ohjelmistot, joilla tiedostomuotoja käsitellään, voidaan jakaa kahteen luokkaan: asiakasohjelmistoihin, joilla loppukäyttäjät voivat lukea tiedostoja, ja pitkäaikaissäilytyksessä käytettyihin ohjelmistoihin, joilla voidaan automaattisesti käsitellä tiedostoja. Pitkäaikaissäilytyksessä käytetyillä ohjelmistoilla voidaan muun muassa tunnistaa tiedostomuotoja, kerätä tiedostoista metatietoja sekä muuntaa tiedostoja yhdestä tiedostomuodosta toiseen. Tärkeitä piirteitä ohjelmistojen kannalta on muun muuassa ohjelmistojen avoimuus, luotettavuus ja tuki. Vapaan lähdekoodin ohjelmistoja voi tutkia eri näkökulmista, joihin kuuluu ohjelmiston dokumentaatio, laaja käyttäjä- ja kehittäjäkanta, joka osallistuu ohjelmiston kehittämiseen ja ylläpitämiseen, sekä ohjelmiston modulaarisuus, joka sallii ohjelmiston käyttämisen esimerkiksi automaattisissa työvoissa [Abe07].

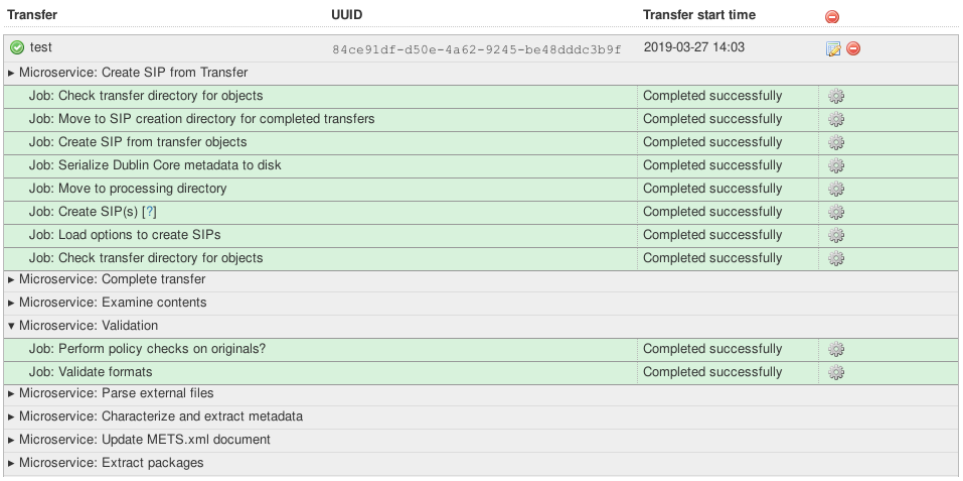
Esimerkiksi toimivien asiakasohjelmistojen puute voi tehdä tiedostomuodosta huonommin soveltuvan pitkäaikaissäilytykseen, vaikka tiedostomuodolle olisi saatavilla validointiin ja tunnistamiseen soveltuvia ohjelmistoja. Toisaalta pitkäaikaissäilytysohjelmistojen puuttuminen tai epävarma toiminta voi vähentää tiedostomuodon soveltuvuutta, vaikka asiakasohjelmistot tukevat tiedostomuotoa kattavasti.

Tämä luku käsittelee tiedostomuotokohtaisesti loppukäyttäjille suunniteltuja asiakasohjelmistoja ja pitkäaikaissäilytykseen soveltuvia ohjelmistoja. Näitä ohjelmistoja arvioidaan ottaen huomioon muun muassa ohjelmistojen avoimuuden, luotettavuuden ja käyttöasteen. Näiden piirteiden perusteella voidaan arvioida tiedostomuotojen soveltuvuutta pitkäaikaissäilytykseen ja mahdollisia ongelmia, jos esimerkiksi tiedostomuodolle ei ole olemassa kattavaa ohjelmistotukea.

5.1 Pitkäaikaissäilytys

Archivematica on mikropalveluarkkitehtuuria hyödyntävä vapaan lähdekoodin ohjelmisto, joka pystyy OAIS-viitemallin mukaisesti hoitamaan aineistojen vastaanottamisen, tarkistamisen ja säilyttämisen. Ohjelmisto on kirjoitettu pääasiassa Python-ohjelmointikieltä käyttäen, ja se on suunniteltu käytettäväksi graafisen web-käyttöliittymän avulla, jolla on mahdollista hoitaa loppukäyttäjän tai ylläpitäjän toimia vaativat toimenpiteet, kuten lähetyspaketin lataaminen palveluun (loppukäyttäjä) ja lähetyspaketin hy-

väksyminen säilytykseen (ylläpitäjä). Kuvassa 5 on esitetty ohjelmiston toimintaa tilanteessa, jossa pitkäaikaissäilytyspalveluun on lähetetty siirtopaketti. Siirtopaketin vastaanottaminen koostuu useista työvaiheista, joiden aikana tiedostot muun muassa tunnistetaan, niiden piirteet kerätään ja niiden sisältö validoitaan.



Kuva 5: Kuvankaappaus Archivematica-ohjelmiston web-käyttöliittymästä. Kuva ilmaisee `test` -nimisen paketin etenemistä vastaanottotyövuossa. Kuvan tämän hetkisessä tilanteessa paketin purkaminen ja validointi on onnistunut, ja paketti on muunnettu OAIS-viitemallin mukaiseksi SIP-siirtopaketiksi.

Tiedostomuodot ja niiden käsittelyyn käytetyt komennot säilytetään *Format Policy Registry* (FPR) -rekisterissä, joka on myös ylläpitäjän muokattavissa [Arc]. Pitkäaikaissäilytyspalvelu voi esimerkiksi lisätä uusia tiedostomuotoja luomalla niiden käsittelyä varten tarvittavat komennot käyttäen Bash- tai Python-ohjelmointikielillä luotuja skriptejä, tai vastaavasti poistaa tiedostomuotoja, joita palvelu ei hyväksy säilytykseen. Nämä komennot voivat vastata esimerkiksi tiedostomuodon tunnistamisesta (*identification*), piirteiden keräämisestä (*characterization*), usean tiedostomuodon muuntaminen yksittäiseen tiedostomuotoon (*normalization*) tai validoinnista.

Archivematica hyödyntää olennusasennuksessa muita ohjelmistoja eri tiedostomuotojen käsittelyssä. Esimerkiksi tiedostomuodon tunnistamista varten Archivematica käyttää FIDO- (*Format Identification for Digital Objects*) ja Siegfried-ohjelmistoja sekä Linux-käyttöjärjestelmien `file`-komentoa. Validointia varten Archivematica käyttää oletuksena ainoastaan JHOVE-ohjelmis-

toa.

JHOVE on tiedostomuotojen tunnistamiseen, validointiin ja piirteiden keräämiseen suunniteltu ohjelmisto. Se on kehitetty Java-ohjelmointikieltä käyttäen, ja se tukee useita tiedostomuotoja kuten aiemmin käsitellyt HTML, TIFF, JPEG ja WAVE -tiedostomuodot. Tuki eri tiedostomuodoille on toteutettu Java-moduuleja käyttäen, minkä ansiosta tuki uusille tiedostomuodoille on mahdollistaa kehittää erillisinä pakkauksina; tämä toisaalta tarkoittaa, että näille tiedostomuodoille ei voi luvata yhtä kattavaa tukea [JHOa]. Esimerkiksi MP3-tiedostomuoto on tuettu erillistä moduulia käyttäen.

JHOVE-ohjelmiston tiedostomuotomoduulit eivät XML-tiedostomuotoa lukuunottamatta ole riippuvaisia erillisistä paketeista tai ohjelmistoista. Moduulit toteuttavat itse tiedostojen lukemisen ja purkamisen käyttämättä erillisiä ohjelmistoja. Käytännössä tämä tarkoittaa, että JHOVE-ohjelmiston käyttäytymiseen eivät vaikuta käyttöjärjestelmään asennetut paketit. Jos esimerkiksi jonkin tiedostomuodon tuki olisi toteutettu erillistä jaettua kirjastoa käyttäen, kaksi eri JHOVE-asennusta voisivat toimia eri tavalla, jos kumpaankin on asennettu sama JHOVE-versio, mutta jaetulla kirjastolla on järjestelmissä eri versiot.

JHOVE on saatavilla vapaana lähdekoodina, ja ohjelmiston bugiseuranta on vapaasti luettavissa GitHub-palvelussa. Näistä ilmenee muun muassa ohjelmistossa olleet ongelmat, niiden ratkaisemiseen kulunut aika ja niitä koskevat keskustelut kehittäjien välillä. Ohjelmisto tukee PDF-tiedostomuotoa, mutta GitHub-palvelussa olevien avoimien bugiraporttien perusteella PDF-tuki sisältää merkittäviä ongelmia ja rajoituksia. JHOVE tarkastaa tiedostot osittain puutteellisesti, ja se voi tämän takia tulkita tiedoston tukeman PDF-profiilin väärin [JHOc]. Ohjelmiston PDF-moduuli pystyy tarkastamaan PDF/A -tiedostojen tekniset piirteet korkealla tasolla, mutta ei ota kantaa piirteisiin, joita on hankalampi selvittää automaattisesti. Kehittäjät suosittelevat tämän takia erillistä *VeraPDF*-validaattoria, joka pystyy tarkistamaan PDF-tiedoston piirteet tarkemmin. Muihin PDF-tiedostomuotoa koskeviin validointiongelmiin kuuluu muun muassa tilanteet, joissa JHOVE voi käyttää useita tunteja yksittäisen PDF-tiedoston käsittelyyn tai kaatua muistin loppumiseen yrittäessään lukea PDF-tiedostoa.

DROID on tiedostomuotojen tunnistamiseen suunniteltu ohjelmisto. Ohjelmisto tunnistaa tiedostomuodon ja tapauksesta riippuen sen versionumeron, mutta ei ota kantaa tiedoston kelvollisuuteen. DROID voi esimerkiksi

tunnistaa PDF/A-1a -muotoiseksi määritellyn asiakirjan, mutta ei pysty selvittämään, täyttääkö asiakirja kyseisen asiakirjastandardin vaatimukset tai onko asiakirja loppukäyttäjän luettavissa.

JHOVE-ohjelmiston tavoin se on kirjoitettu Java-ohjelmointikieltä käyttäen, mutta käyttää tiedostomuotojen tunnistamiseen Java-moduulien sijasta XML-tiedostomuotoa käyttävää PRONOM-tietokantaa. Kyseisen tietokannan avulla tiedostomuoto voidaan selvittää etsimällä tiedoston bittivirrasta tiettyjä sarjoja ("signature") ja/tai lukemalla tiedostonimen pääte. DROID ei siis käsittele tiedostojen sisäistä rakennetta, mikä tekee tiedostomuotojen tunnistamisesta nopeaa. Toisaalta, toisiaan muistuttavat tiedostomuodot voi olla vaikeampi tunnistaa. Esimerkiksi XML ja HTML käyttävät hyvin samankaltaista syntaksia. Lisäksi tiedostomuodon tunnistaminen voi olla hankalaa tai jopa mahdotonta, jos tiedostomuodon selvittäminen vaatii tiedoston rakenteen lukemista. Esimerkiksi PDF-asiakirjoissa asiakirjan versionumero on luettavissa PDF-tiedoston bittivirran alusta (esimerkiksi `%PDF-1.3`), mutta PDF-standardissa 1.4 ja sitä uudemmissa versioissa versionumero voidaan tallentaa asiakirjan sisälle omaan tietueeseen [DRObj]. Jos versionumerot poikkeavat, PDF-asiakirjaan tallennettu tietue sisältää oikean arvon. Tässä tapauksessa asiakirjan PDF-versionumeron lukeminen vaatisi PDF-tiedostomuotoa tukevan kirjaston käyttöä, mitä DROID ei tue.

```
<InternalSignature ID="23" Specificity="Specific">
    <ByteSequence Reference="BOFoffset">
        <SubSequence MinFragLength="0" Position="1"
            SubSeqMaxOffset="0" SubSeqMinOffset="0">
            <Sequence>255044462D312E33</Sequence>
            <DefaultShift>9</DefaultShift>
            <Shift Byte="25">8</Shift>
            <Shift Byte="2D">4</Shift>
            <Shift Byte="2E">2</Shift>
            <Shift Byte="31">3</Shift>
            <Shift Byte="33">1</Shift>
            <Shift Byte="44">6</Shift>
            <Shift Byte="46">5</Shift>
            <Shift Byte="50">7</Shift>
        </SubSequence>
    </ByteSequence>
</InternalSignature>
```

Listaus 2: Ote PDF 1.3 -versiota koskevasta sarjasta. Sarjan perusteella tiedoston alussa tulee olla merkkijono `%PDF-1.3` (heksadesimaalimuodossa `255044462D312E33`).

5.2 Asiakasohjelmistot

Aiemmin käsitellyille PDF/A, HTML ja ODF -tiedostomuodoille on saatavilla useita tiedostomuotojen käsittelyyn soveltuvia kirjastoja. Näille tiedostomuodoille on myös olemassa määritelmät, joiden avulla on mahdollista toteuttaa tiedostomuodon lukemiseen kykeneviä ohjelmistoja. Lisäksi on olemassa testejä, joilla ohjelmiston yhteensopivuus voidaan tarkastaa automatisoidusti. Nämä testit muun muassa tarkistavat käsitteleekö tiedostomuotoa käsittelevä ohjelmisto oikein eri reunatapaukset, ja ne voivat olla ohjelmistokehityksessä hyödyllisempiä kuin pelkkä tiedostomuodon rakenteen ja toiminnallisuuden kuvaava tekninen asiakirja.

HTML-standardille on W3C-järjestön kehittämä testikokoelma [Web]. Kyseinen järjestö on vastuussa myös itse HTML-standardin kehityksestä. PDF/A-1 -standardille on vastaavasti olemassa *Isartor Test Suite* -nimellä tes-

tikokoelma, jonka suunnittelusta ja kehityksestä on vastannut *PDF/A Competence Center* [Isa]. Kyseinen järjestö koostuu *PDF Association* -järjestön piiriin kuuluvista vapaaehtoisista työntekijöistä, mutta myös muutamista ISO-toimikuntaan kuuluvista jäsenistä, jotka ovat osallistuneet ISO-standardin määrittelyyn. Aiemmista tiedostomuodoista poiketen ODF-tiedostomuodolle ei ole olemassa muodollista testikokoelmaa. *Apache Software Foundation* on kuitenkin suunnitellut testikokoelman luomista [Cor].

Asiakirjojen käsitellessä on tärkeätä tiedostomuodon lukemisen lisäksi, että sen sisältö voidaan toistaa samalla tavoin riippumatta ohjelmistosta. HTML-tiedostojen rakenteen lukemiseen ja käsittelemiseen on olemassa monia ohjelmistoja kuten *libxml2*, *html5lib* ja *HtmlUnit*. Nämä ohjelmistot eivät kuitenkaan vastaa sisällön esittämisestä käyttäjälle, vaan ovat ainoastaan yksi osa HTML-asiakirjojen esittämiseen tarvittavaa kokonaisuutta. Tavallisesti asiakirjojen esittämiseen käytetään web-selainta. Esimerkiksi HTML-standardi edellyttää että web-selaimet esittävät saman HTML-asiakirjan samalla tavoin. Tämä ei kuitenkaan ole aina mahdollista johtuen esimerkiksi web-selaimien tukemista ominaisuuksista ja toteutuskohtaisista eroista. Yhteensopivuutta voi kuitenkin parantaa rajaamalla HTML-asiakirjassa käytettyjä ominaisuuksia: esimerkiksi JavaScript-skriptikielen käyttö voidaan kieltää kokonaan asiakirjoissa, jotka eivät sisällä interaktiivisia elementtejä. Esimerkiksi Uuden-Seelannin kansallinen kirjasto käytti WordStar-tiedostomuodon muunnoksessa uudemman HTML 5 -standardin sijasta vanhempaa mutta laajemmin tuettua HTML 4 -standardia [GM14].

HTML-asiakirjojen esittämiseen käytetään web-selainta, joka puolestaan hyödyntää jotakin selainmoottoria [ABG+16]. Selainmoottorin vastuulla on kääntää HTML-muodossa tallennettu tiedosto käyttäjälle näytettävään visuaaliseen muotoon. Selainmoottoreihin kuuluu muun muassa *Mozilla Firefox* -selaimessa käytetty *Gecko*, *Chromium* -selaimessa ja moniin siihin pohjautuvissa selaimissa (esimerkiksi *Google Chrome* ja *Opera*) käytetty *Blink*, sekä *Safari* -selaimessa käytetty *WebKit*. Kaikki näistä selainmoottoreista on julkaistu avoimena lähdekoodina, mikä edesauttaa selainmoottorien uudelleenkäyttöä muissa tilanteissa: esimerkiksi *WebKit* -selainmoottori on käytössä *Electron* -ohjelmistokehyksessä, jonka avulla voidaan luoda web-teknologioita hyödyntäviä ja järjestelmäriippumattomia työpöytäsovelluksia.

PDF-asiakirjojen esittäminen tapahtuu yleensä erillisillä PDF-lukijoilla, mutta myös web-selaimet voivat sisältää tuen PDF-tiedostojen lukemiselle.

PDF-lukijoihin kuuluu muun muassa *Adobe* -yrityksen kehittämä kaupallinen suljetun lähdekoodin *Adobe Reader*, sekä avoimen lähdekoodin *Poppler*-kirjasto jota useat PDF-lukijat käyttävät PDF-tiedostojen esittämiseen [Pop]. Lisäksi web-selaimilla on mahdollista lukea PDF-tiedostoja käyttäen web-selainten JavaScript-tukea hyödyntävää *PDF.js* -kirjastoa [PDF]. Esimerkiksi *Mozilla Firefox* sisältää oletuksena PDF-lukutuen kyseistä kirjastoa käyttäen, ja *Google Chrome*:en kyseinen tuki on saatavilla erillisen web-selainliitännäisen avulla.

ODF-tekstiasiakirjat (*OpenDocument*) on luettavissa useita tekstinmuokkausohjelmia käyttäen mukaan lukien *Microsoft*-yrityksen kehittämä kaupallinen *Microsoft Office* -toimisto-ohjelmistopaketti ja vapaan lähdekoodin *LibreOffice* -toimisto-ohjelmistopaketti. Huomattava ero näiden ohjelmistojen välillä on tiedostomuodolle annettu tuki: *LibreOffice* käyttää ODF-tiedostomuotoa oletuksena, ja *Microsoft Office* yhtenä vaihtoehtona ohjelmistopaketin oletuksena käyttämän `DOCX`-tiedostomuodon lisäksi. OpenDocument-tiedostomuoto on myös käytössä eri maiden hallituksissa. Esimerkiksi Euroopan unioni suosittelee instituutioiden käyttävän joko ODF- tai OOXML-tiedostomuotoa muokattavien tekstiasiakirjojen tallentamiseen [Sig].

Aiemmin käsitellyistä tiedostomuodoista erityisesti JPEG on laajassa käytössä: suurin osa laitteista ja ohjelmistoista osaa lukea ja/tai luoda JPEG-tiedostoja. JPEG-tiedostomuotoa käsitteleviin ohjelmistokirjastoihin kuuluvat muun muassa vuonna 1991 julkaistu *libjpeg*, kyseiseen kirjastoon perustuva *libjpeg-turbo* ja Googlen itsenäisesti kehittämä *Guetzli* [AOS+17]. Nämä kirjastot eroavat toisistaan esimerkiksi käyttötarkoitusten, suorituskyvyn ja tiedostokoon optimoinnin kannalta. Esimerkiksi *libjpeg-turbo* hyödyntää SIMD-käskyjä JPEG-tiedostojen luomisen ja purkamisen nopeuttamiseen, ja *Guetzli* keskittyy pienempiin tiedostokokoihin säilyttäen kuitenkin samannäköisen kuvan hyödyntäen malleja ihmisnäöstä.

JPEG-tiedostomuodosta poiketen *JPEG 2000* ei ole yhtä laajassa käytössä, mutta ohjelmistoista esimerkiksi validointisovellus *JHOVE* ja kuvien ja videoiden käsittelyyn soveltuva *ffmpeg* pystyvät lukemaan JP2-tiedostoja. JP2-tiedostoja lukeviin kirjastoihin kuuluvat muun muassa *OpenJPEG* sekä *Grok*. Sen sijaan web-selaimissa ei ole vielä laajaa tukea JP2-tiedostomuodolle [JPEa]. TIFF-kuvatiedostojen lukemiseen ja kirjoittamiseen on saatavilla avoimen lähdekoodin `libtiff`-niminen kirjasto. Kyseinen kirjasto ei ennen vuotta 2004 sisältänyt oletuksena tukea LZW-pakkausmenetelmäl-

le, vaan LZW-tuki oli saatavilla erillisen lähdekoodipaketin avulla [LZW]. LZW-menetelmää koskevat patentit raukesivat vuonna 2004 ja `libtiff` -kirjastoon palautettiin LZW-tuki versiossa 3.7.

EXIF (*Exchangeable image file format*) on TIFF-pohjainen standardi, joka määrittelee yhtenäisen metatietoformaatin kuva- ja äänitiedostoihin. TIFF ja JPEG tukevat tätä standardia, mutta toteuttavat ne eri tavoin: esimerkiksi JPEG-tiedostoihin kaikki EXIF-metatieto tallennetaan yhden sovelluskohtaisen tunnuksen (`APP1`) alle. EXIF-standardin mukaisesti JPEG-tiedostoissa `APP1` -tunnus tulee liittää tiedoston alkuun [EA02]. Tämä edesauttaa tiedoston johdonmukaisuutta sijoittamalla kaiken metatiedon tiedoston alkuun mutta toisaalta aiheuttaa ristiriidan toisen metatietostandardin kanssa: JFIF-standardi käyttää JPEG-tiedostoissa `APP0` -tunnusta ja vaatii EXIF-standardin tavoin, että kyseinen tunnus sijoitetaan tiedoston alkuun [JPEb]. Tämä ristiriita on ratkaistu useimmissa JPEG-tiedostoissa sijoittamalla JFIF-tunnus tiedoston alkuun ennen EXIF-tunnusta; uudemmat JPEG-lukijat pystyvät tulkitsemaan tällaiset tiedostot, mutta vanhemmat JFIF- ja EXIF-lukijat eivät välttämättä pysty lukemaan kyseisiä tiedostoja. TIFF-standardi sen sijaan ei määrittele metatietotunnuksille järjestystä, vaan ne voidaan sijoittaa vapaasti eri puolille tiedoston bittivirtaa. JPEG 2000 ei kuitenkaan tue EXIF-standardia vaan hyödyntää erilaista XML-pohjaista metadatastandardia [SCE01].

Kuvatiedostojen sisäinen metatieto voi sisältää esimerkiksi kuvan resoluution, digitaalikameran teknisiä tietoja. Tämän lisäksi metatietoja voi käyttää myös muihin käyttötarkoituksiin: esimerkiksi mikroskoopeilla otettuihin kuviin on suositeltu lisättävän metatietoja, joista ilmenee missä olosuhteissa kuva on otettu [LRA+10]. Tässä tapauksessa kuvatiedoston mielivaltainen metatietotuki on hyödyksi pitkäaikaissäilytyksen kannalta, sillä kuvatiedostoa koskevat semanttiset tiedot on mahdollista säilyttää itse kuvatiedostossa eikä esimerkiksi pelkästään erillisenä METS-asiakirjana.

EXIF-metatietoja on mahdollista muokata ja lukea käyttäen *ExifTool* -nimistä sovellusta. Nimestään poiketen sovellus ei tue ainoastaan EXIF-standardia vaan pystyy käsittelemään muita metatietostandardeja kuten *XMP* ja *IPTC* sekä digitaalikameroiden valmistajien omia metatietorakenteita, joita ei ole virallisesti dokumentoitu [Toe15]. Kyseinen sovellus on käytettävissä komentorivillä ja erillisen ohjelmointirajapinnan avulla, minkä ansiosta se soveltuu pitkäaikaissäilytyksen kaltaisiin digitaalisiin työvoihin.

FLAC on vapaa tiedostomuoto, ja sen toistamiseen ja kirjoittamiseen vaaditut kirjastot ovat saatavilla avoimena lähdekoodina *libFLAC*-paketissa [Usi]. Tähän kirjastoon kuuluu myös testikokoelma, joka varmistaa muun muassa, että äänitiedosto on mahdollista muuntaa FLAC-tiedostomuotoon ilman, että ääni-informaatio muuttuu [FLAa]. Muihin FLAC-tiedostomuotoa lukeviin kirjastoihin kuuluvat *Flake* ja *FLACCL*.

FLAC-tiedostoja on mahdollista käyttää useilla vapaasti saatavilla ohjelmistoilla. Näihin ohjelmistoihin kuuluu muun muassa äänentoisto-ohjelmistoja sekä sovelluksia, joilla voi luoda tai käsitellä FLAC-tiedostoja. Kyseisillä ohjelmistoilla voi muun muassa muokata tiedostojen metatietoja tai muuntaa äänitiedostoja yhdestä tiedostomuodosta toiseen; tämä voi olla hyödyllistä siirtäessä muita häviöttömiä äänitiedostoja FLAC-muotoon. Lisäksi käyttöjärjestelmät kuten Windows 10, Android ja macOS pystyvät toistamaan oletuksena FLAC-tiedostoja.

WAVE-äänitiedostot ovat laajassa käytössä erityisesti äänenmuokkaussovelluksissa kuten avoimen lähdekoodin *Audacity*-sovelluksessa. WAVE-tiedostomuotoon perustuva *Broadcast Wave* -tiedostomuoto on käytössä järjestelmissä, joissa äänitietoa halutaan siirtää järjestelmien välillä ilman riskiä laadun tai informaation menettämisestä [Bro]. WAV-tiedostoille on myös kattava tuki käyttöjärjestelmissä kuten *Microsoft Windows*, *macOS* ja *Linux*.

MP3-tiedostomuoto on laajassa käytössä ja toimii useissa äänisovelluksissa ja käyttöjärjestelmissä. Voimassaolevat patentit rajoittivat tiedostomuodon käyttöä avoimen lähdekoodin projekteissa ennen vuotta 2017, mutta äänitiedostomuodon lukemiseen ja luontiin oli saatavilla myös tätä ennen useita avoimen lähdekoodin kirjastoja. Näihin kirjastoihin kuuluu muun muassa MP3-tiedostojen luontiin käytetty *LAME*, ja tiedostojen lukemiseen käytetty *mpg123*. Patentti ei rajoittanut lähdekoodimuodossa olevien ohjelmistojen levittämistä, mutta esimerkiksi eri sovelluksissa ei ollut oletuksena MP3-tukea, vaan loppukäyttäjän täytyi itse ladata ja asentaa MP3-yhteensopivuuden lisäävä jaettu kirjasto. Esimerkiksi *Audacity* -äänenmuokkaussovellus vaatii erillisen LAME-koodauskirjaston MP3-tiedostojen käsittelyä varten [Aud]. Huomioitavaa on kuitenkin, että ohjelmistopatentit eivät rajoittaneet MP3-tiedostomuotoa käsittelevän lähdekoodin kehittämistä ja vapaata jakamista vaan ainoastaan lähdekoodista käännettyjen ohjelmistojen levittämistä. LAME-kehittäjät itse totesivat, että lähdekoodin levittäminen ei ole lainvas-

taista ja ottivat esimerkiksi ISO-standardit, jotka sisältävät lähdekoodia, joiden tarkoitus on havainnollistaa standardien mahdollista toteutusta [LAM].

Kaikilla mainituilla äänitiedostomuodoilla on myös kattava web-selaintuki lukuunottamatta *Internet Explorer* -selainta, jonka uusin versio tukee mainituista tiedostomuodoista vain MP3-äänitiedostoja. Kaikille tiedostomuodoille on saatavilla alustariippumattomat avoimen lähdekoodin kirjastot, ja tiedostomuodoille on kattava tuki myös käyttöjärjestelmissä. Huomattavaa MP3-tiedostomuodossa on se, että ohjelmistopatentit eivät ole rajoittaneet avoimen lähdekoodin projektien syntymistä. Näiden projektien käyttö voi kuitenkin tehdä pitkäaikaissäilytyksestä vastuussa olevista järjestöistä korvausvelvollisia.

6 Tarkistuslista

Tämä luku käsittelee aiemmin esiteltyjen ja tutkittujen piirteiden kannalta kehitettyä tarkistuslistaa, esitellen tarkistuslistan eri kehitysvaiheet aloittaen tarkistuslistan ensimmäisestä versiosta ja päättyen asiantuntijoiden palautteen perusteella luotuun lopulliseen versioon tarkistuslistasta. Tarkistuslistan lopullinen versio löytyy gradun liitteestä A. Tarkistuslista koostuu kysymyksistä, joiden avulla tiedostomuodon piirteitä voidaan arvioida pitkäaikaissäilytyksen kannalta. Nämä piirteet eivät kuitenkaan ole yksiselitteisiä ja voivat riippua pitkäaikaissäilytyspalvelun toiminnasta ja vaatimuksista, kuten pitkäaikaissäilytykseen saatavilla olevasta tallennustilasta. Piirteet voivat myös vaatia tutkimustyötä tiedostomuotoa pitkäaikaissäilytykseen ottavalta taholta, etenkin jos tiedostomuoto ei ole laajassa käytössä.

Tarkistuslistan tarkoitus on auttaa tiedostomuotojen säilytyskelpoisuuden arvioinnissa ja välttää tilanteita, joissa aiemmin säilytykseen otettu tiedostomuoto osoittautuukin epäkelvolliseksi jonkin seikan takia. Esimerkiksi Uuden-Seelannin kansallinen kirjasto kehitti muunnostyökalun WordStar-tiedostomuodolle, kun ilmeni, että tiedostomuotoa tukevia ohjelmistoja ei ole enää saatavilla nykyisin käytössä oleville käyttöjärjestelmille [GM14].

Tarkistuslista on myös hyödyllinen, kun harkitaan säilytettäväksi tiedostomuotoa, joka ei ole laajassa käytössä. Käytännön esimerkkinä voi olla esimerkiksi aineistolaji, joka halutaan säilyttää, mutta jolle ei ole määritelty tiedostomuotoa. Tämä ongelma ilmeni esimerkiksi *Harvard University Libraries* -hankkeessa, jossa suunniteltiin sähköpostiviestien pitkäaikaissäilytystä [GG10]. Sähköpostiviestien lähettämiselle ja vastaanottamiselle on olemassa laajassa käytössä olevat standardit, mutta sähköpostiviestin tallentaminen vaatii standardin luomista ja vastaavasti tiedostomuodon valitsemista tai suunnittelemista säilytettävää materiaalia varten. Tämän kaltaisessa tapauksessa tarkistuslista voi osoittautua hyväksi työkaluksi eri tiedostomuotojen arvioinnissa.

Luku käsittelee ensin tarkistuslistan ensimmäiseen versioon valittuja piirteitä, piirteiden perusteella luotuja kysymyksiä ja perusteluja näiden kysymyksien valinnalle. Tarkistuslistan ensimmäisestä versiosta on kerätty palautetta Suomen kansallisen pitkäaikaissäilytyspalvelun kehittäjiltä, jonka perusteella tarkistuslistasta on hiottu lopullinen versio. Tarkistuslistan lopullinen versio löytyy tutkielman lopusta liitteenä A. Tämän lopullisen

tarkistuslistan käyttöä on havainnollistettu luvun lopussa esimerkillä, jossa tarkistuslistaa käytetään kahden tiedostomuodon arviointiin.

6.1 Piirteet säilytyksen kannalta

Digitaaliset kopionsuojausmenetelmät

Digitaaliset kopionsuojausmenetelmät voivat haitata tiedostomuodon käyttöä pitkäaikaissäilytyksessä [HS14]. Tämä voi tarkoittaa esimerkiksi salasanasuojausta, jolla tiedoston sisältämä informaatio salataan. Kyseessä voi myös olla ulkopuolisia tunnistuspalvelimia hyödyntävä kopionsuojaus. Tässä tapauksessa aineistoa on mahdollista käyttää normaalisti niin kauan, kuin kopionsuojauksesta vastaavat palvelut ovat toiminnassa.

Jos digitaalisia kopionsuojausmenetelmiä käyttäviä aineistoja on otettu pitkäaikaissäilytettäväksi eikä virhettä ole huomattu ajoissa, aineisto voi muuttua täysin lukukelvottomaksi, jos salausta ei voi enää purkaa. Näin voi tapahtua esimerkiksi, jos suojauksesta vastuussa oleva oikeudenhaltija ei ole enää tavoitettavissa.

Digitaalisia kopionsuojausmenetelmiä on mahdollista tunnistaa erilaisten työkalujen avulla [HS14]. Tämä kuitenkin vaatii ylimääräistä kehitystyötä ja sisältää vaaran, että ohjelmistovirheen takia tiedosto otetaan säilytykseen puutteellisen kopionsuojausmenetelmän tunnistuksen vuoksi. Pitkäaikaissäilytyspalvelua ylläpitävän tahon tulisi siis suosia tiedostomuotoja, jotka eivät tue laisinkaan digitaalisia kopionsuojausmenetelmiä.

Avoimen lähdekoodin ohjelmistojen saatavuus

Avoimen lähdekoodin ohjelmistot edesauttavat tiedostomuodon käyttöä pitkäaikaissäilytyksessä. Tämän voi huomata monista tuotantokäytössä olevista pitkäaikaissäilytysprojekteista, jotka käyttävät järjestelmien kehityksessä runsaasti avoimen lähdekoodin komponentteja. iPRES-konferenssien vuonna 2010-2018 julkaistuista artikkeleista löytyy mainintoja useista eri pitkäaikaissäilytykseen soveltuvista ohjelmistoista. Kymmenen mainituimman ohjelmiston lista sisältää kahdeksan vapaan lähdekoodin alla julkaistua ohjelmistoa ja kaksi kaupallista suljetun lähdekoodin ohjelmistoa.

Tiedostomuotokohtaisissa ohjelmistoissa avoimen lähdekoodin ohjelmistoista on hyötyä erityisesti, jos tiedostomuodon käsittely halutaan lisätä osaksi palvelun olemassaolevaa työvuota. Lähdekoodi helpottaa tässä ta-

pauksessa ohjelmiston kääntämistä uudelle alustalle ja pitkäaikaissäilytystä koskevien ominaisuuksien lisäämistä.

Avoimen lähdekoodin ohjelmistojen kehitys

Vaikka tiedostomuodolle on saatavilla avoimen lähdekoodin alainen ohjelmisto, lähdekoodin avoimuus itsessään ei kuitenkaan itsessään ole riittävä peruste esimerkiksi tietyn ohjelmiston käyttämisessä. Ohjelmiston tulee myös olla laadukas ja aktiivisessa kehityksessä. Ohjelmiston laatua tulisi siten arvioida eri mittareilla ennen käyttöönottoa: näihin mittareihin kuuluu muun muassa kehitysyhteisö (edesauttaa nopeaa kehitystä ja vikojen korjausta), kattava dokumentaatio ja koodin modulaarisuus (edesauttaa ohjelmiston käyttöönottoa) [Abe07]. Ohjelmiston kehityksen tulisi myös olla vakaalla pohjalla; aiemmin käsitelty JHOVE2-ohjelmisto oli kehityksen alla ainakin vuodesta 2009 lähtien, mutta sen kehitys hidastui ja lopulta lakkasi kokonaan rahoituksen päätyttyä vuonna 2011 [JHOe]. Ohjelmistoa harkitessa voisi ottaa huomioon muun muassa ohjelmiston kehitystä tukevat järjestöt ja yritykset, sekä itse kehitystyöhön osallistuvat osapuolet ja kehitystyön aktiivisuuden [CAH03]. Aktiivisuudella ei tarkoiteta ainoastaan kuinka usein lähdekoodiin tehdään muutoksia, vaan siinä voi ottaa huomioon myös projektia koskevat bugiraportit ja kuinka pitkään niiden käsittely kestää. Huomioitavaa on, että bugiraporttien määrä ei tarkoita, että ohjelmisto olisi huonolaatuinen, vaan se voi viestiä myös aktiivisesta käyttäjäyhteisöstä.

Vaikka avoimen lähdekoodin ohjelmisto ei ole kehityksen alla, se ei kuitenkaan estä ohjelmiston ottamista käyttöön. Tämä kuitenkin edellyttää, että pitkäaikaissäilytyksestä vastaava taho myös ottaa vastuun ohjelmiston jatkokehityksestä. Riippuen tiedostomuodon monimutkaisuudesta, ohjelmisto voi pysyä käyttökelpoisena pitkään vaatimatta jatkokehitystä.

Standardointi

Avoimen lähdekoodin lisäksi tiedostomuodosta tulisi olla standardoitu siten, että standardin tai määritelmän perusteella on mahdollista luoda ohjelmisto kyseisen tiedostomuodon käsittelyyn. Määritelmän lisäksi kehityksessä olisi hyödyksi kattava testikokoelma tiedostoja, joiden avulla voidaan varmistaa automatisoidusti että ohjelmistot pystyvät avaamaan kaikki kelvolliset testitiedostot ja vastaavasti hylkäämään epäkelvollisiksi määritellyt testitie-

dostot. Standardin tulisi olla myös yksiselitteinen, sillä standardissa olevat epäselvyydet voivat johtaa ohjelmistokohtaisiin eroihin ja pahimmassa tapauksessa siihen, ettei yhdellä ohjelmistolla luotua tiedostoa voi avata toisella ohjelmistolla, vaikka kummatkin ohjelmistot tukevat samaa tiedostomuotoa. Epäselvien piirteiden löytäminen ei ole kuitenkaan itsestään selvää ja voi vaatia esimerkiksi oma-aloitteista tutkimustyötä. Esimerkiksi PDF-asiakirjan käyttäjälle esitetty sivu saattaa näyttää erilaiselta järjestelmään asennetuista kirjasintyypeistä riippuen. Tämän takia pitkäaikaissäilytyksessä käytetyissä PDF/A -asiakirjoissa kirjasintyypit pitää myös sisällyttää asiakirjaan [ISO05].

Toisaalta aiemmin käsitellyistä tiedostomuodoista esimerkiksi TIFF on suunniteltu modulaariseksi: kuvainformaatio on mahdollista pakata monilla eri pakkausmenetelmillä ja tiedostoon on mahdollista lisätä mielivaltaisesti jäsenneltyä metatietoa. Nämä seikat voivat johtaa helposti yhteensopivuusongelmiin, jos ohjelmistot eivät tue samoja ominaisuuksia. Tätä varten TIFF-standardiin lisättiin versiossa 6 erillinen Baseline TIFF -määritelmä, joka määrittelee ominaisuudet, joita kaikkien TIFF-lukijoiden tulee tukea [TIF]. Näihin kuuluu muun muassa sallitut pakkausmenetelmät ja kuvatiedoston pakolliset tietueet. Baseline TIFF ei ole kuitenkaan ainoa TIFF-tiedostomuotoa hyödyntävä määritelmä, vaan tiedostomuotoja tunnistava JHOVE-ohjelmisto tukee muita julkisia profiileja (public profile), jotka sallivat TIFF-tiedostomuodon käytön erilaisissa käyttötapauksissa: näistä profiileista esimerkiksi *GeoTIFF* soveltuu maantieteellisten töiden kuten karttojen säilyttämiseen [RRG+00].

Aineiston määrä

Kansallinen pitkäaikaissäilytyspalvelu määrittelee säilytys- ja siirtokelpoiset tiedostomuodot [PASb]. Siirtokelpoiset tiedostomuodot viittaavat tiedostomuotoihin, jotka voidaan ottaa pitkäaikaissäilytykseen. Näihin kuuluvat myös tiedostomuodot, jotka eivät sellaisenaan sovellu pitkäaikaissäilytykseen, mutta jotka voidaan ottaa vastaan pitkäaikaissäilytyspalvelussa.

Jos aineistoa omistavalla taholla on runsaasti pitkäaikaissäilytykseen siirrettäviä aineistoja, on luultavasti tarkoituksenmukaisempaa ottaa aineistot vastaan välittömästi ja suunnitella tiedostomuodon muunnos myöhemmin. Tässä tapauksessa tiedostomuodon muuntaminen voi vaatia erillisen muunnostyökalun kehitystyötä, erityisesti jos kyseessä on uusi tiedostomuoto, joka

on vain muutaman ryhmän käytössä. Tämän kehitystyön määrän arvioiminen voi olla kuitenkin vaikeaa, eikä ole taattua, että pitkäaikaissäilytyspalvelulla tai muulla taholla on resursseja sen toteuttamiseksi. Tästä huolimatta aineistosta on kuitenkin parempi säilyttää ainakin jotain kuin ei mitään.

Pakkausmenetelmän häviöttömyys/häviöllisyys

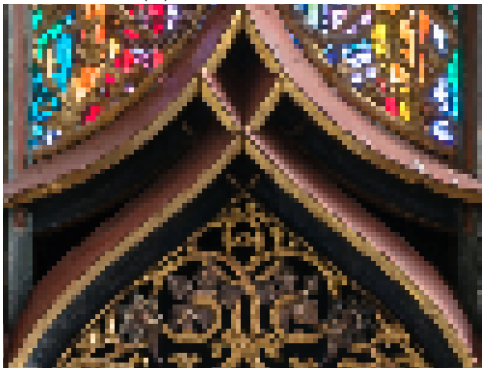
Aiemmin käsitellyissä tiedostomuodoissa oli mukana sekä häviöllisiä että häviöttömiä pakkausmenetelmiä hyödyntäviä tiedostomuotostandardeja. Pitkäaikaissäilytyksessä pyritään säilyttämään alkuperäinen informaatio parhaalla mahdollisella laadulla, jolloin käytettäisiin ensisijaisesti häviötöntä tiedostomuotoa. Tämä pätee silloin, kun käytössä on riittävän paljon tallennustilaa, jolloin rajoitteeksi voi muodostua esimerkiksi taideteoksen skannaukseen käytetyn laitteen tarkkuus.

Tiedostojen säilytyksessä on kuitenkin myös mahdollista käyttää häviöllisiä tiedostomuotoja, jolloin on mahdollista vähentää tiedostojen säilyttämiseen kuuluvia kustannuksia säilyttäessä tiedosto tarvittavalla kuvan laadulla. Jos tiedoston koolle on määritelty yläraja, häviöllisen pakkausmenetelmän käyttö sallii käytännössä suuremman informaatiomäärän tallentamisen tiedostoon huolimatta häviöllisestä pakkausmenetelmästä. Tätä on havainnollistettu PNG ja JPEG -tiedostomuodoilla kuvassa 6. Esimerkiksi skannattaessa fyysistä asiakirjaa tai kuvaa on todennäköisesti mielekkäämpää valita korkea skannaustarkkuus, ja tallentaa kuva häviölliseen tiedostomuotoon, kuin valita alhaisempi skannaustarkkuus, ja tallentaa kuva häviöttömään tiedostomuotoon. Toisaalta häviöllistä pakkausmenetelmää käyttäessä käyttäjä ei voi tarkoin määritellä, miten tiedoston sisältämät yksityiskohdat tulee käsitellä; esimerkiksi JPEG-kuvatiedostoa pakatessa käyttäjä määrittelee yksittäisen laatuarvon (*quality*), jonka perusteella sovellus hoitaa koko JPEG-tiedoston pakkauksen. Käyttäjä ei voi siis vaikuttaa kuvan yksittäisiin osa-alueisiin. Jos esimerkiksi kuvatiedostoa ei ole jostain syystä saatavilla korkealla resoluutiolla ja pakkausmenetelmän käyttö voi hävittää tärkeitä yksityiskohtia, häviöttömän tiedostomuodon käyttö on luultavasti tärkempää. Tällaisia kuvia voisivat olla esimerkiksi lääketieteelliset kuvat.

Häviöttömät pakkausmenetelmät kuitenkin tekevät siirron tiedostomuotojen välillä helpoksi. Tämä on hyödyllistä esimerkiksi, jos erittäin korkean tiedostokoon videotiedostosta halutaan tehdä pienempi loppukäyttäjälle soveltuva kopio. Tälloin uudet kopiot voidaan aina luoda alkuperäisestä hä-

(a) alkuperäinen

(b) JPG (häviöllinen)



(c) PNG (häviötön)

Kuva 6: Kuvasarjassa sarjan ensimmäinen kuva *a* on tallennettu häviölliseen JPEG -tiedostomuotoon (kuva *b*) ja häviöttömään PNG -tiedostomuotoon (kuva *c*). Kummassakin tapauksessa kuvan enimmäistiedostokoko on rajoitettu viiteen megatavuun; JPEG-tiedostoa tallentaessa tiedoston koko pudotetaan alle viiden megatavun lisäämällä pakkauksen määrää, kun PNG-tiedoston kohdalla kuvan resoluutiota vähennetään, kunnes pakkaamaton kuva mahtuu viiteen megatavuun. Käytännössä JPEG-tiedosto säilyttää kuvan laadun paremmin huolimatta pakkauksen määrästä. PNG-tiedoston resoluutiota täytyy puolestaan pudottaa huomattavasti ennen kuin kuva mahtuu viiteen megatavuun. Tämän perusteella on järkevämpää laittaa pitkäaikaissäilytykseen korkean resoluution sisältävä häviöllinen kuva kuin alhaisemman resoluution sisältävä häviötön kuva, jos kuvalle on määritelty enimmäistiedostokoko. Kuvasarjassa käsitelty alkuperäiskuva on julkaistu *Creative Commons Attribution-Share Alike 4.0 International* -lisenssin alaisena Wikimedia-sivustolla: `https://commons.wikimedia.org/wiki/File:Pfarrwerfen_Kirche_Innenraum_01.jpg`

viöttömästä tiedostosta ilman vaaraa informaation liiallisesta vähenemisestä. Häviöttömän tiedostomuodon käyttö on myös hyödyksi tehdessä muunnosta tiedostomuodosta toiseen. Esimerkiksi FLAC-tiedostomuodon kirjaston sisältämä työkalu `flac` mahdollistaa vanhan ja uuden tiedoston vertaamisen muunnoksen aikana. Muunnettaessa esimerkiksi WAV-tiedostoa FLAC-tiedostomuotoon on mahdollista tarkistaa, että kummankin tiedoston sisältämä informaatio on muunnoksen päätteeksi identtinen ja varmistaa siten, että tiedostomuodon muunnos on onnistunut [FLAa]. Tiedostomuotojen muunnos voi tulla aiheelliseksi, jos uusi tiedostomuoto pystyy pakkaamaan informaation tehokkaammin tai jos tiedostomuoto sisältää muita piirteitä, jotka tekevät pitkäaikaissäilytetyn tiedoston muunnoksesta kannattavan operaation (esimerkiksi huomattavasti parempi suorituskyky tai ohjelmistotuki). Esimerkiksi Iso-Britannian kansalliskirjaston teettämän tutkimuksen mukaan häviöttömien TIFF-värikuvien pakkaussuhde oli huomattavasti parempi käytettäessä ZIP-pakkausmenetelmää LZW-menetelmän sijasta [MD16].

Aiemmin käsitelty *Archivematica* -ohjelmisto sisältää vapaaehtoisten toiminnallisuuden tiedostomuotojen normalisoinnille (normalization); ohjelmiston mukana tulevia perusasetuksia käyttäen esimerkiksi kaikkien videotiedostojen sisältämät video- ja äänivirrat voidaan kopioida `mkv`-videosäiliömuotoisiin tiedostoihin ja täten tallentaa kaikki pitkäaikaissäilytetyt videotiedostot yhtä tiedostomuotoa käyttäen. Säiliömuodosta riippuen tämä voi edellyttää informaation purkamista ja uudelleenpakkaamista, jos kohdetiedostomuoto ei esimerkiksi tue tiettyä pakkausmenetelmää; jos lähdetiedosto käyttää jo häviötöntä pakkausmenetelmää, vaaraa informaation häviämisestä ei ole.

Tekniset rajoitukset

Tiedostomuodoissa voi myös piileä erilaisia teknisiä rajoituksia, jotka on aiheellista huomioida, etenkin jos pitkäaikaissäilytykseen lähetettyjä tiedostoja aiotaan muuntaa toiseen tiedostomuotoon esimerkiksi normalisoinnin tai loppukäyttäjäkopioiden luomisen yhteydessä. Yleisin näistä lienee tiedoston enimmäiskoko: esimerkiksi TIFF-kuvatiedostojen ja WAVE-äänitiedostojen enimmäistiedostokoko on noin neljä gigatavua[4] [TIF] [WAV]. Tämä voi esimerkiksi rajoittaa pakkaamattomien äänitiedostojen pituutta tai kuvatiedostojen enimmäisresoluutiota. Tiedoston pitkäaikaissäilytykseen lähettävä taho

[4]Kumpikin tiedostomuoto käyttää tiedostokoon kuvaamiseen 32-bittistä etumerkitöntä kokonaislukua, mikä rajoittaa tiedostojen enimmäiskoon 4,294,967,296 tavuun

voi tässä tapauksessa joko pilkkoa aineiston useaksi tiedostoksi tai lähettää tiedoston pitkäaikaissäilytykseen tiedostomuodossa, jossa ei ilmene samaa teknistä rajoitusta.

Muihin teknisiin seikkoihin voi kuulua esimerkiksi kuvatiedostoissa käytetty värisyvyys. Sekä TIFF ja JPEG -kuvatiedostot tallentavat kaksiulotteiset värikuvat RGB-mallia käyttäen, jossa jokaista kuvan pikseliä kohden on olemassa punaisen, vihreän ja sinisen värin sävyä kuvaava numeerinen arvo. Tämän numeerisen arvon tarkkuus voi kuitenkin vaihdella kuvatiedostosta riippuen: esimerkiksi JPEG-kuvatiedostot tukevat 8-bittisiä ja 12-bittisiä värisävyarvoja [Wal92]. TIFF-kuvatiedostot sen sijaan tukevat sekä 8-bittisiä että 16-bittisiä värisävyarvoja [TIF]. Useimmat käytössä olevat kuluttajatuotteet kuten digitaalikamerat ja tietokonenäytöt tukevat ainoastaan 8-bittisiä värisävyjä, mutta 10-bittiset värisävyt ovat yleistymässä pikkuhiljaa esimerkiksi televisioissa ja tietokonenäytöissä [LZT+14]. Ammattilaiskäytössä korkea värisävytarkkuus on hyödyksi erityisesti kuvien tai videoiden jälkikäsittelyssä. Esimerkiksi *Adobe Photoshop* -kuvanmuokkausohjelmasta tukee 8-bittisen värisyvyyden lisäksi myös 16-bittisiä ja 32-bittisiä värikanavia sisältäviä kuvia [Pho]. Pitkäaikaissäilytyksen kannalta tämä tarkoittaa, että vaikka kuvatiedosto muunnetaan häviöttömästä tiedostomuodosta toiseen häviöttömään tiedostomuotoon, kuvan informaatiossa voi tapahtua hävikkiä, ellei kumpikin tiedostomuoto tue samaa värisyvyyttä.

Ohjelmistopatentit

Tutkielman luvussa 3 käsiteltiin ohjelmistopatentteja ja niistä mahdollisesti ilmeneviä ongelmia pitkäaikaissäilytyksen kannalta. MP3-tiedostomuodon kohdalla ohjelmistopatentit eivät kuitenkaan vaikuttaneet avoimen lähdekoodin ohjelmistojen saatavuuteen. Ohjelmistopatenttiin perustuvat ohjelmistot voivat kuitenkin asettaa pitkäaikaissäilytyksestä vastuussa olevan tahon tai aineistoa paketoivan tahon korvausvelvollisuuden alaiseksi riippuen maakohtaisista laista [Wil12]. Pitkäaikaissäilytyspalvelun kannattaa siis ensisijaisesti suosia avoimia tiedostomuotoja, jos palvelusta vastuussa oleva taho ei voi olla varma palvelun mahdollisesti patentteja rikkovista seikoista.

Tunnistettavuus

Tiedostomuodon ja sen mahdollisen version tunnistamisen tulisi olla helppoa ja johdonmukaista; käytännössä tiedostomuoto pitäisi pystyä tunnistamaan esimerkiksi tiedoston otsakkeesta sijaitsevasta merkkijonosta ilman, että tiedostomuotoa tunnistavan sovelluksen tarvitsee tulkita tiedoston sisältämää tietomallia. Pitkäaikaissäilytysjärjestelmät kuten aiemmin käsitelty *Archivematica* aloittavat yksittäisen tiedoston käsittelyn selvittämällä sen tiedostomuodon, minkä jälkeen tiedoston käsittely ohjataan tiedostomuotoa parhaiten käsittelevälle sovellukselle: käsittelystä vastuussa oleva sovellus voi olla laajassa käytössä oleva sovellus tai sisäisesti kehitetty sovellus, joka keskittyy sellaisen tiedostomuodon validointiin, jolle ei ole saatavilla vartavasten kehitettyä validointiohjelmistoa. Tiedoston jatkokäsittely voidaan myös tarpeen mukaan hajauttaa tiedostomuodon mukaan eri järjestelmiin. Esimerkiksi jotkin kuva- tai äänitiedostomuodot voidaan esimerkiksi muuntaa tiettyyn tiedostomuodoon ennen pitkäaikaissäilytystä, mikä voi vaatia tiedostoa käsittelevältä koneelta enemmän keskusmuistia ja/tai laskentatehoa kuin pelkkään validointiin keskittynyt kone. Tiedostomuodon tunnistaminen ennen jatkokäsittelyä edesauttaa tämän toiminnallisen vaatimuksen toteutumista.

Tiedostomuodon tai sen version väärä tunnistaminen voi joko keskeyttää tiedoston säilytysprosessin kokonaan tai johtaa virheelliseen validointiin: PDF/A-tiedostojen tapauksessa väärin tunnistettu versio voi tarkoittaa, että tiedostoa ei tarkisteta tarpeeksi kattavasti tai että tiedoston validointi epäonnistuu, vaikka tiedosto oli todellisuudessa kelvollinen [Kli17]. Vastaavasti HTML5-asiakirjojen syntaksi näyttää monilta piirteiltään samanlaiselta aiempaan HTML4-asiakirjoihin verrattuna, mutta HTML5 poistaa kielestä aiemmin tuettuja elementtejä ja myös lisää uusia elementtejä [HTMa]. Tämän takia on suositeltavaa, että HTML-asiakirjan versiot on erotettavissa toisistaan; HTML-asiakirjoissa tämä tapahtuu käyttämällä tiedoston alussa sijaitsevaa `<!DOCTYPE>` -otsaketta. Käytännössä web-selainmoottorit pystyvät ainakin jossain määrin käsittelemään HTML-asiakirjoja, joissa versionumeroa ei ole määritelty tai joissa kummankin standardin elementtejä on käytetty yhdessä asiakirjassa. Ongelmaksi muodostuu kuitenkin web-selainmoottoreiden yksittäiset erot virheellisten asiakirjojen esittämisessä, mikä voi näkyä esimerkiksi asiakirjan sommittelun hajoamisena.

6.2 Arviointi

Kehitysryhmän palaute

Tarkistuslistan ensimmäisen version käyttökelpoisuuden arvioimiseksi tarkistuslistasta kerättiin palautetta CSC:n pitkäaikaissäilytyspalvelun kehittäjiltä. Palautteen laativat kolmen henkilön ryhmä, jonka jäsenet ovat vastuussa muun muassa palvelun suunnittelusta ja ohjelmistokehityksestä. Ryhmä ehdotti muutoksia muutamaan tarkistuslistan kysymykseen ja myös otti esille aiheesta aiemmin tehtyjä selvityksiä ja raportteja.

Digitaalisia kopionsuojausmenetelmiä koskevassa kysymyksessä pidettiin oleellisempana kopionsuojauksen olemassaoloa vastaanotettavassa tiedostossa eikä sen mahdollista tukea käytettävässä tiedostomuodossa. Tämä tarkoittaisi että digitaalisten kopionsuojausmenetelmien hylkääminen tulisi painottua vastaanottovaiheeseen eikä niinkään tiedostomuotoa arvioitaessa. Digitaalisten kopionsuojausmenetelmien välttäminen on kuitenkin suositeltavaa

Aineiston määrää ja muunnettavuutta koskevaa taulukko arvioitiin puutteelliseksi: aineiston heterogeenisyys on myös merkittävä ongelma, joka voi ilmentyä määrältään pienenä aineistona, joka voi kuitenkin osoittautua monimutkaisemmaksi kuin määrältään iso aineisto ja vaatia enemmän kehitystyötä, ennen kuin koko aineisto on mahdollista ottaa pitkäaikaissäilytykseen.

Pakkausmenetelmän häviöttömyys/häviöllisyys arvioitiin kompromissiksi tallennustilan ja laadun välillä: kansainvälisissä pitkäaikaissäilytyspiireissä vältetään häviöllisten pakkausmenetelmien käyttöä. Pakkausmenetelmissä tulisi siten suosia joko häviöttömiä pakkausmenetelmiä tai olla käyttämättä pakkausta laisinkaan. Häviöttömistä pakkausmenetelmistä esille otettiin LZW-pakkausmenetelmä, jota käytetään muun muassa TIFF-kuvatiedostoissa ja jonka avulla on aineistosta riippuen mahdollista saavuttaa lähes yhtä korkea pakkaussuhde kuin häviöllistä pakkausta käyttävillä JPEG-kuvatiedostoilla. Palautteen mukaan suurin osa kansainvälisistä pitkäaikaissäilytyspalveluista asettaa aineiston laadun ensisijaiseksi tekijäksi riippumatta vaadittavasta tallennustilasta, mikä vähentää kysymyksen tarpeellisuutta.

Teknisiä rajoituksia koskeva kysymys arvioitiin hyvin tulkinnanvaraiseksi ja laajaksi alueeksi, jonka arviointi on vaikeaa. Teknisten rajoitusten sijasta ehdotettiin tiedostomuodon kompleksisuuden arviointia, jossa selvitetään tiedostomuodon ominaisuuksien määrä ja suositaan ensisijaisesti yksinkertaisempia tiedostomuotoja.

Aikaisemmat selvitykset

Palautteessa otettiin esille myös raportteja ja selvityksiä, joissa on eri tavoin arvioitu tiedostomuotoja pitkäaikaissäilytyksessä. Näihin kuuluvat Yhdysvaltain kongressinkirjaston selvitys [Sus], Iso-Britannian kansallisarkiston raportti [Bro08], Alankomaiden kuninkaallisen kirjaston selvitys [RVW08] sekä InterPARES 2 -projektin raportti [McL07].

Raportit käsittelevät tiedostomuotojen piirteitä, mutta ne sisältävät eroja muun muassa käsitellyissä piirteissä ja määrittelyissä. Esimerkiksi InterPARES 2 -raporttiin kuuluu pitkäaikaissäilytyspalveluja koskevia ehdotuksia, jotka eivät koske tiedostomuotojen arviointia [McL07]. Näihin ehdotuksiin kuuluu esimerkiksi yhteistyö muiden pitkäaikaissäilytyspalvelujen kanssa, jotka hyväksyvät säilytykseen samoja tiedostomuotoja.

Eniten tarkistuslistaa muistuttaa Alankomaiden kuninkaallisen kirjaston selvitys, joka käyttää pisteytysjärjestelmää tiedostomuodon eri piirteiden arviointiin [RVW08]. Pisteytysjärjestelmän avulla tiedostomuodolle lasketaan pistemäärä asteikolla 0-100, jonka perusteella tiedostomuotoja on mahdollista vertailla keskenään. Pisteytysjärjestelmässä esiintyvät piirteet ja niiden tärkeysaste ovat kuitenkin subjektiivisia: raportin mukaan pääpaino kirjaston pääpaino on aineistojen lukukelpoisuuden säilyttämisessä ja piirteiden tärkeys voi riippua palvelukohtaisesti. Esimerkiksi metatietotuki voi olla tärkeämpi, jos aineisto halutaan säilyttää myös muokkauskelvollisena. Tämä voi olla aiheellinen kysymys, jos esimerkiksi tekstiasiakirjasta halutaan säilyttää muokkauskelpoinen tiedosto, johon sisältyy myös asiakirjan versiohistoria. Säilytettävät aineistot voivat myös vaihdella projektikohtaisesti. Esimerkiksi jotkin tieteelliset aineistot on luotu vartavasten kehitetyillä ohjelmistoilla, jotka ovat vähässä käytössä tieteellisen yhteisön ulkopuolella. Tämä ei välttämättä estä tiedostomuodon ottamista pitkäaikaissäilytykseen, jos kyseistä tiedostomuotoa käsittelevät ohjelmistot ovat vapaasti saatavilla ja tiedostomuoto on kattavasti dokumentoitu.

Raporteissa on esillä samoja piirteitä kuin tarkistuslistassa. Esimerkiksi avoimet tiedostomuodot luokiteltiin kaikissa raporteissa pitkäaikaissäilytyksen kannalta tärkeäksi piirteeksi. Näistä piirteistä tiedostomuodon avoimuus, digitaalisten kopionsuojausmenetelmien puuttuminen ja avoimen lähdekoodin ohjelmistojen saatavuus koettiin tärkeiksi. Raporteissa otettiin esille myös tarkistuslistan ulkopuolisia piirteitä, kuten tiedostomuodon kestävyys (kyky tunnistaa esimerkiksi tiedoston sisällön korruptoitumista), monimutkaisuus

ja ihmisluettavuus.

Yhdysvaltojen kongressinkirjaston, InterPARES 2 -projektin ja Alankomaiden kirjaston raporteissa oli samankaltaiset ohjeet pakkausmenetelmien käyttöön: aineisto tulee tallettaa joko häviötöntä pakkausmenetelmää käyttäen tai täysin pakkaamattomana [McL07]. InterPARES 2 -raportin mukaan suurin osa selvitykseen osallistuneista organisaatioista hyväksyy säilytykseen vain pakkaamattomia tai häviötöntä pakkausta käyttäviä tiedostoja. Näistä organisaatioista neljä hyväksyy säilytykseen vain pakkaamattomia tiedostoja.

Pakkaamattomien tiedostojen säilyttämistä perusteltiin Yhdysvaltojen kongressinkirjaston selvityksessä ihmisluettavuutta häiritsevänä tekijänä sekä mahdollisena ongelmana, jos pakkausmenetelmä on patentin alainen [Sus]. Pakkausmenetelmän käyttö on kuitenkin tarpeellista, kun säilytettävänä on aineistoa, joka vie pakkaamattomana reilusti tallennustilaa ja joka on mahdollista pakata tehokkaasti aineiston piirteitä hyödyntäen: näihin aineistoihin voi kuulua esimerkiksi korkearesoluutioisia ääni- ja videotiedostoja. Pakkausmenetelmät tulisi tässä tapauksessa toteuttaa laajasti käytettyjä ja vapaasti saatavilla olevia algoritmeja käyttäen.

Raporteissa ei käsitellä teknisiä rajoituksia, jotka voivat esimerkiksi esiintyä tiedostokokorajoituksina. Sen sijaan metatietotuki otetaan esille tärkeänä piirteenä esimerkiksi kuvan väriavaruuden kuvaamiseen [Sus]. Kattava metatietotuki tarkoittaisi, että esimerkiksi kuvatiedostojen värisyvyys ja sen muutokset voidaan tallentaa metatietoihin. Tallennetut metatiedot voivat kuvata esimerkiksi tiettyä ohjelmistoa ja muunnoksessa käytettyjä asetuksia. Tiedostokokorajoitusten kohdalla metatiedon avulla voidaan säilyttää viittaukset alkuperäiseen aineistoon, esimerkiksi jos äänitiedosto täytyy pilkkoa useaksi tiedostoksi teknisten rajoitusten vuoksi. Koska teknisten rajoitusten määrittäminen on hankalaa, on tärkeämpi keskittyä sen sijaan kattavaan metatietotukeen tiedostomuotoja arvioitaessa.

Raporteissa ja tarkistuslistaa koskevassa palautteessa otetaan esille digitaaliset kopionsuojausmenetelmät. Kopionsuojausmenetelmät eivät kuitenkaan ole este tiedostomuodon hyväksymiselle, jos ne voidaan tunnistaa vastaanottovaiheessa. Oleellista on kopionsuojausmenetelmien käyttö yksittäisissä tiedostoissa eikä sitä koskeva vapaaehtoinen tiedostomuototuki. Tämä seikka ilmenee Alankomaiden kirjaston raportissa ja Yhdysvaltojen kongressinkirjaston selvityksessä [RVW08] [Sus].

Tarkistuslistasta saadun palautteen perusteella tarkistuslistan kysymyk-

siä on tarkennettu ja niiden tärkeysjärjestystä muutettu. Tarkistuslistan teknisiä rajoitteita koskeva kysymys on poistettu, sillä se arvioitiin vaikeasti arvioitavaksi piirteeksi. Tarkistuslistaan lisättiin tiedostomuodon metatietotukea koskeva kysymys. Metatiedot mahdollistavat sekä tiedostoa koskevien teknisten piirteiden tallentamisen että tiedoston muunnoksia koskevien tietojen tallentamisen.

Digitaalisia kopiosuojausmenetelmiä ja pakkausmenetelmiä koskevia kysymyksiä on siirretty tärkeysjärjestyksessä alemmas. Oleellisinta on välttää kopionsuojausmenetelmien käyttöä vastaanotetussa aineistossa. Tiedostomuodon vapaaehtoinen kopionsuojaustuki ei siis estä tiedostomuodon käyttöönottoa.

6.3 Tarkistuslistan käyttö

Tarkistuslistan käyttöä havainnollistetaan arvioimalla kahta tiedostomuotoa: PNG-kuvatiedostomuoto (*Portable Network Graphics*) ja AIFF-äänitiedostomuoto (*Audio Interchange File Format*). Tutkittavat tiedostomuodot tallentavat aiemmin käsiteltyjen tiedostomuotojen tavoin kuva- ja äänitietoa. Tiedostomuodoiksi valittiin laajassa käytössä olevat tiedostomuodot: esimerkiksi PNG-kuvat ovat laajasti tuettuja web-selaimissa [Can]. Tarkistuslistan kysymystä *Aineiston määrä ja muunnettavuus* ei käsitellä, sillä se riippuu tarkistuslistaa käyttävän organisaation tarpeista. Kysymys *Pakkausmenetelmän häviöttömyys/häviöllisyys* riippuu vastaavasti organisaation tarpeista, joten kysymyksen kohdalla otetaan esille vain tiedostomuodon tukemat pakkausmenetelmät.

Portable Network Graphics, PNG

Avoimen lähdekoodin ohjelmistojen saatavuus

	Useita avoimen lähdekoodin ohjelmistoja: lukemista ja kirjoittamista tukeva *libpng* [libc] ja *LodePNG* [Lod]. Laaja webselaintuki [Can].

Avoimen lähdekoodin ohjelmiston kehitys

	Viimeisin versio libpng-kirjastosta julkaistu vuonna 2019 [libc]. Tiedostomuodon kehityksestä on vastuussa W3C [PNGc]. libpng-kirjaston kehitykseen on osallistunut yksittäisten kehittäjien lisäksi myös *Google* ja *Arm Holdings* [libd].

Digitaaliset kopionsuojausmenetelmät

	Tiedostomuoto ei tue digitaalisia kopionsuojausmenetelmiä [PNGc].

Standardointi

	Tiedostomuodosta on saatavilla W3C-järjestön määritelmä [PNGc]. Tiedostomuoto on myös ISO-standardoitu [ISO04b] (ISO/IEC 15948-1:2004). Tiedostomuodolle on saatavilla tiedostoista koostuva testikokoelma [Pngb].

Metatietotuki

	Tiedostomuoto tukee XMP-määritelmän mukaisia metatietoja [Exi]. Määritelmä sallii ennaltamäärättyjen tietuiden lisäksi mielivaltaisten tietuiden lisäämisen, minkä avulla voidaan tallentaa tarkempia tietoja tiedostosta.

Tunnistettavuus

	PNG-tiedostot on tunnistettavissa otsakesarjan perusteella (`89 50 4e 47 0d 0a 1a 0a`) [PNGc]. Otsakkeen perusteella ei voi kuitenkaan määritellä tiedostomuodon tarkkaa versiota (1.0, 1.1 tai 1.2).

Pakkausmenetelmän häviöttömyys/häviöllisyys

	PNG-tiedostomuoto hyödyntää häviötöntä DEFLATE-pakkausmenetelmää [PNGc].

Ohjelmistopatentit

	PNG-tiedostomuoto suunniteltiin patenttivapaaksi [PNGa]. Kehitysryhmän mukaan tiedostomuodon toteuttavat ohjelmistot eivät ole patenttien alaisia.

Audio Interchange File Format, AIFF

Avoimen lähdekoodin ohjelmistojen saatavuus

	Useita avoimen lähdekoodin ohjelmistoja: lukemista ja kirjoittamista tukeva *LibAiff* [Libb], *aiff* [aifc] ja *libavformat* [liba]. Kattava käyttöjärjestelmätuki (macOS [App], Microsoft Windows [Med]).

Avoimen lähdekoodin ohjelmiston kehitys

	libavformat ja aiff -kirjastot ovat aktiivisen kehityksen alla, ja kummastakin projektista on julkaistu uusi versio vuoden 2019 aikana [liba] [aifc]. Tiedostomuotoa koskeva standardi on ainoastaan Apple-yrityksen määrittelemä [AIFb].

Digitaaliset kopionsuojausmenetelmät

	Tiedostomuoto ei tue digitaalisia kopionsuojausmenetelmiä [AIFb].

Standardointi

	Tiedostomuodosta on saatavilla Applen kirjoittama standardi [AIFb]. Tiedostomuodolle ei ole saatavilla testikokoelmaa.

Metatietotuki

	AIFF-tiedostomuoto sallii metatiedon lisäämisen ylimääräisiä lohkoja käyttäen (esimerkiksi *Name Chunk*) [AIFb]. Tiedostoihin voi lisätä mielivaltaista metatietoa *Application Chunk* -lohkoja hyödyntäen.

Tunnistettavuus

	AIFF-tiedostot on tunnistettavissa otsakesarjan perusteella (`AIFF`, heksadesimaalimuodossa `46 4F 52 4D 00`) [AIFb]. AIFF ja siihen perustuva AIFF-C -tiedostomuoto ovat erotettavissa toisistaan tiedoston alusta löytyvän toisen otsakesarjan perusteella (`AIFF` tai `AIFC`). AIFF-standardista on olemassa kolme versiota (1.1, 1.2 ja 1.3), jotka ovat kuitenkin teknisiltä seikoiltaan identtiset, lukuunottamatta *Apple ProDOS* -käyttöjärjestelmää koskevaa virhettä, joka korjattiin versiossa 1.3. Version tunnistaminen ei siis ole tarpeellista.

Pakkausmenetelmän häviöttömyys/häviöllisyys

	AIFF-tiedostomuoto ei käytä pakkausmenetelmiä [AIFb].

Ohjelmistopatentit

	AIFF-tiedostomuoto ei ole Yhdysvaltojen kongressinkirjaston teettämän tutkimuksen perusteella patenttien alainen [AIFa].

Esimerkit havainnollistavat tarkistuslistan käyttöä tiedostomuodon arvioinnissa. Esimerkiksi pakkausmenetelmiä ja tunnistettavuutta koskevat kysymykset on mahdollista selvittää pelkästään tiedostomuodon määritelmää käyttäen. Avoimen lähdekoodin ohjelmistoja ja ohjelmistopatentteja koskevat kysymykset vaativat ylimääräistä tutkimustyötä esimerkiksi avoimen lähdekoodin ohjelmistojen selvittämisessä. Esitetyissä tapauksissa avoimen lähdekoodin ohjelmistoja koskeva kysymys oli mahdollista ratkaista etsimällä tiedostomuotoa tukevia avoimen lähdekoodin ohjelmistoja, ja selvittämällä, mitä kirjastoja ohjelmistot käyttävät tiedostomuototuen takaamiseksi. Tarkistuslistan kysymykset on siis ratkaistavissa vapaasti saatavilla olevia tietolähteitä käyttäen.

7 Johtopäätökset

Tutkielmassa esiteltiin pitkäaikaissäilytyksen tavoitteet, perusperiaatteet ja onnistuneen pitkäaikaissäilytyksen työvaiheet aineiston lähettämisestä sen arkistointiin asti. Tämän lisäksi käsiteltiin tilanteita, joissa aineiston pitkäaikaissäilytys epäonnistuu muun muassa tiedostossa olevien puutteiden tai rajoitusten vuoksi.

Tiedostomuodoista löydettyjen piirteiden ja ongelmatilanteiden perusteella luotiin tarkistuslista, joka mahdollistaa tiedostomuodon arvioinnin pitkäaikaissäilytystarpeisiin. Tarkistuslistasta kerättiin palautetta Suomen kansallisen pitkäaikaissäilytyspalvelua kehittävältä ryhmältä, joiden palautteen avulla arvioitiin tarkistuslistan käyttökelpoisuutta ja hiottiin siinä esiintyneitä kohtia ja niiden tärkeysjärjestystä. Tavoitteena oli luoda helppokäyttöinen tarkistuslista, joka toimii suuntaa antavana viitekehyksenä tiedostomuodon arviointiin ja vähentää siihen kuluvaa aikaa. Tutkimuksessa käytettyjen kirjallisuuslähteiden ja tarkistuslistasta saadun palautteen perusteella on selvää, että tiedostomuodoissa on hyvin paljon piirteitä, joiden tärkeys voi riippua tilannekohtaisesti. Nämä tilanteet voivat koskea esimerkiksi säilytettävän aineiston muotoa (kuva- vai äänitiedosto?) tai organisaation omista tarpeista (saatavilla olevat kehitysresurssit). Tarkistuslista soveltuu kuitenkin tiedostomuotojen arviointiin siten, että tiedostomuoto voidaan joko hylätä, tai siitä voidaan tehdä jatkotutkimusta tiedostomuodon lopullista käyttöönottoa ajatellen. Jatkotutkimus voi tässä tapauksessa tarkoittaa esimerkiksi mahdollisesti pitkäaikaissäilytettävien tiedostojen tutkimista olemassa olevilla ohjelmistoilla tai varta vasten kehitetyillä prototyypeillä. Koska jatkotutkimus vaatii enemmän työtä ja aikaa, on tärkeää, että tiedostomuoto voidaan hylätä aikaisessa vaiheessa turhan työmäärän vähentämiseksi.

Lähteet

[Abe07] Aberdour, Mark: *Achieving quality in open-source software.* IEEE software, 24(1), 2007.

[ABG+16] Anderson, Brian, Bergstrom, Lars, Goregaokar, Manish, Matthews, Josh, McAllister, Keegan, Moffitt, Jack ja Sapin, Simon: *Engineering the servo web browser engine using Rust.* Teokses-

sa *Proceedings of the 38th International Conference on Software Engineering Companion*, sivut 81–89. ACM, 2016.

[AIFa] *AIFF (Audio Interchange File Format)*. https://www.loc.gov/preservation/digital/formats/fdd/fdd000005.shtml, vierailtu 2019-04-23.

[AIFb] *Audio Interchange File Format: "AIFF- Version 1.3*. http://www-mmsp.ece.mcgill.ca/Documents/AudioFormats/AIFF/Docs/AIFF-1.3.pdf, vierailtu 2019-04-23.

[aifc] *go-audio/aiff: Battle tested aiff decoder/encoder*. https://github.com/go-audio/aiff, vierailtu 2019-04-23.

[AOS+17] Alakuijala, Jyrki, Obryk, Robert, Stoliarchuk, Ostap, Szabadka, Zoltan, Vandevenne, Lode ja Wassenberg, Jan: *Guetzli: Perceptually Guided JPEG Encoder*. arXiv preprint arXiv:1703.04421, 2017.

[App] *aiff - AVFileType | Apple Developer Documentation*. https://developer.apple.com/documentation/avfoundation/avfiletype/1386870-aiff, vierailtu 2019-04-23.

[Arc] *Preservation Planning | Documentation (Archivematica 1.8) | Archivematica: open-source digital preservation system*. https://www.archivematica.org/en/docs/archivematica-1.8/user-manual/preservation/preservation-planning/, vierailtu 2018-12-04.

[Arc18] *Archivematica: open-source digital preservation system*, 2018. https://www.archivematica.org/en/, vierailtu 2018-10-08.

[Aud] *LAME Legal Issues - Audacity Manual*. https://alphamanual.audacityteam.org/m/index.php?title=LAME_Legal_Issues&oldid=57455, vierailtu 2019-02-03.

[Bit] *BitCurator*. https://bitcurator.net/bitcurator/, vierailtu 2018-11-04.

[Bro] *Specification of the Broadcast Wave Format (BWF).* https://tech.ebu.ch/docs/tech/tech3285.pdf, vierailtu 2019-02-04.

[Bro08] Brown, Adrian: *Digital preservation guidance note 1: Selecting file formats for long-term preservation.* The National Archives, Reino Unido, sivut 1–10, 2008. https://www.nationalarchives.gov.uk/documents/selecting-file-formats.pdf.

[C+08] Committee, PREMIS Editorial *et al.*: *Premis data dictionary for preservation metadata.* Preservation, 2008.

[CAH03] Crowston, Kevin, Annabi, Hala ja Howison, James: *Defining open source software project success.* ICIS 2003 Proceedings, sivu 28, 2003.

[Can] *Can I use... Support tables for HTML5, CSS3, etc.* https://caniuse.com/#search=png, vierailtu 2019-04-23.

[COP] *COPTR.* https://coptr.digipres.org/Main_Page, vierailtu 2018-11-04.

[Cor] *ODF - Corinthia - Apache Software Foundation.* https://cwiki.apache.org/confluence/display/Corinthia/ODF, vierailtu 2019-01-14.

[DROa] *File profiling tool (DROID) - The National Archives.* http://www.nationalarchives.gov.uk/information-management/manage-information/policy-process/digital-continuity/file-profiling-tool-droid/, vierailtu 2018-11-04.

[DROb] *PDF version numbers based on deprecated mechanism.* https://github.com/digital-preservation/droid/issues/114, vierailtu 2019-01-28.

[EA02] Electronics, Japan ja Association, Information Technology Industries: *JEITA CP-3451, Exchangeable Image File Format for Digital Still Cameras: Exif Version 2.2.* 2002. http://www.exif.org/Exif2-2.PDF, vierailtu 2019-01-14.

[ERLS16] Espenschied, Dragan, Rechert, Klaus, Liebetraut, Thomas ja Stobbe, Oleg: *Exhibiting Digital Art via Emulation.* Teokses-

sa *Proceedings of the 13th International Conference on Digital Preservation*, 2016.

[Exi] *The Metadata in PNG files - Exiv2*. https://dev.exiv2.org/projects/exiv2/wiki/The_Metadata_in_PNG_files, vierailtu 2019-04-23.

[Fed] *Multimedia/MP3 - Fedora Project Wiki*. https://fedoraproject.org/w/index.php?title=Multimedia/MP3&oldid=492768, vierailtu 2019-01-01.

[FLAa] *FLAC - faq*. https://xiph.org/flac/faq.html, vierailtu 2019-02-05.

[FLAb] *FLAC - features*. https://xiph.org/flac/features.html, vierailtu 2018-12-10.

[FLAc] *FLAC - format*. https://xiph.org/flac/format.html, vierailtu 2019-05-04.

[GG10] Goethals, Andrea ja Gogel, Wendy: *Reshaping The Repository: The Challenge Of Email Archiving*. Teoksessa *Proceedings of the 7th International Conference on Digital Preservation*, 2010.

[GM14] Gattuso, Jay ja McKinney, Peter: *Converting WordStar to HTML4*. Teoksessa *Proceedings of the 11th International Conference on Digital Preservation*, 2014. https://phaidra.univie.ac.at/detail_object/o:378066, vierailtu 2018-11-01.

[HS14] Hein, Stefan ja Steinke, Tobias: *DRM and digital preservation: A use case at the German National Library*. Teoksessa *Proceedings of the 11th International Conference on Digital Preservation*, 2014. https://phaidra.univie.ac.at/detail_object/o:378066, vierailtu 2018-11-01.

[HTMa] *HTML5 Differences from HTML4*. https://www.w3.org/TR/html5-diff, vierailtu 2019-02-15.

[HTMb] *Using HTML sections and outlines - Developer guides | MDN*. https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Using_HTML_sections_and_outlines, vierailtu 2018-11-21.

[ICC] *Image technology colour management - Architecture, profile format and data structure.* `http://www.color.org/specification/ICC1v43_2010-12.pdf`, vierailtu 2018-11-27.

[Isa] *Isartor Test Suite - PDF Association.* `https://www.pdfa.org/isartor-test-suite/`, vierailtu 2019-01-07.

[ISO93] ISO: *Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 3: Audio.* ISO 11172`3:1993, International Organization for Standardization, 1993.

[ISO94] ISO: *Information technology – Digital compression and coding of continuous-tone still images: Requirements and guidelines.* ISO 10918`1:1994, International Organization for Standardization, 1994.

[ISO00] ISO: *Information technology – Document description and processing languages – HyperText Markup Language (HTML).* ISO 15445`1:2000, International Organization for Standardization, 2000.

[ISO01] ISO: *Electronic still-picture imaging – Removable memory – Part 2: TIFF/EP image data format.* ISO 12234`2:2001, International Organization for Standardization, 2001.

[ISO04a] ISO: *Graphic technology – Prepress digital data exchange – Tag image file format for image technology (TIFF/IT).* ISO 12639`2004, International Organization for Standardization, 2004.

[ISO04b] ISO: *Information technology – Computer graphics and image processing – Portable Network Graphics (PNG): Functional specification.* ISO 15948`1:2004, International Organization for Standardization, 2004.

[ISO05] ISO: *Document management – Electronic document file format for long-term preservation – Part 1: Use of PDF 1.4 (PDF/A-1).* ISO 19005`1:2005, International Organization for Standardization, 2005.

[ISO15] ISO: *Information technology – Open Document Format for Office Applications (OpenDocument) v1.2 – Part 1: OpenDocument Schema*. ISO 26300`1:2015, International Organization for Standardization, 2015.

[JHOa] *JHOVE | Documentation*. `http://jhove.openpreservation.org/documentation/`, vierailtu 2019-01-02.

[JHOb] *JHOVE | JSTOR/Harvard Object Validation Environment*. `http://jhove.openpreservation.org/`, vierailtu 2018-11-04.

[JHOc] *JHOVE reporting PDF as v1.3 and as ISO PDF/A-1, Level B · Issue 101 · openpreserve/jhove*. `https://github.com/openpreserve/jhove/issues/101`, vierailtu 2019-01-23.

[JHOd] *jhove2 - Bitbucket*. `https://bitbucket.org/jhove2/`, vierailtu 2018-11-04.

[JHOe] *JHOVE2 | Digital Curation Center*. `http://www.dcc.ac.uk/resources/external/jhove2`, vierailtu 2018-11-04.

[JPEa] *Can I use... Support tables for HTML5, CSS3, etc*. `https://caniuse.com/#feat=jpeg2000`, vierailtu 2018-12-04.

[JPEb] *The Metadata in JPEG files - Exiv2*. `http://dev.exiv2.org/projects/exiv2/wiki/The_Metadata_in_JPEG_files`, vierailtu 2019-01-14.

[Kli17] Klindt, Marco: *PDF/A considered harmful for digital preservation*. Teoksessa *Proceedings of the 14th International Conference on Digital Preservation*, 2017. `https://ipres2017.jp/wp-content/uploads/15Marco-Klindt.pdf`, vierailtu 2018-11-03.

[LAM] *LAME Technical FAQ*. `http://lame.sourceforge.net/tech-FAQ.txt`, vierailtu 2019-02-15.

[Lav00] Lavoie, Brian: *Meeting the challenges of digital preservation: The OAIS reference model*. OClC Newsletter, 243:26–30, 2000.

[LHK+15] Lehtonen, Juha, Helin, Heikki, Koivunen, Kimmo, Lehtonen, Kuisma ja Tiainen, Mikko: *A National Preservation Solution for*

Cultural Heritage. Teoksessa *Proceedings of the 12th International Conference on Digital Preservation*, 2015. https://phaidra.univie.ac.at/view/o:429524, vierailtu 2018-11-01.

[liba] *FFmpeg/libavformat at master - FFmpeg/FFmpeg.* https://github.com/FFmpeg/FFmpeg/tree/master/libavformat, vierailtu 2019-04-23.

[Libb] *The LibAiff Library.* http://aifftools.sourceforge.net/libaiff/, vierailtu 2019-04-23.

[libc] *libpng Home Page.* http://www.libpng.org/pub/png/libpng.html, vierailtu 2019-04-23.

[libd] *PNG REFERENCE LIBRARY AUTHORS.* http://www.libpng.org/pub/png/src/libpng-AUTHORS.txt, vierailtu 2019-04-23.

[Lod] *LodePNG.* https://lodev.org/lodepng/, vierailtu 2019-04-23.

[LRA+10] Linkert, Melissa, Rueden, Curtis T, Allan, Chris, Burel, Jean Marie, Moore, Will, Patterson, Andrew, Loranger, Brian, Moore, Josh, Neves, Carlos, MacDonald, Donald *et al.*: *Metadata matters: access to image data in the real world.* The Journal of cell biology, 189(5):777–782, 2010. http://jcb.rupress.org/content/jcb/189/5/777.full.pdf.

[LZT+14] Liu, M., Zhai, G., Tan, S., Zhang, Z., Gu, K. ja Yang, X.: *HDR2014 - A high dynamic range image quality database.* Teoksessa *2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, sivut 1–6, July 2014.

[LZW] *Bringing back LZW compression | Linux.com | The source for Linux information.* https://www.linux.com/news/bringing-back-lzw-compression, vierailtu 2019-01-16.

[McL07] McLellan, Evelyn Peters: *Selecting Digital File Formats for Long-Term Preservation: InterPARES 2 Project General Study 11 Final Report*, 2007. http://www.interpares.org/display_file.cfm?doc=ip2_file_formats(complete).pdf.

[MD16] May, Peter ja Davies, Kevin: *Practical Analysis of TIFF File Size Reductions Achievable Through Compression.* Teoksessa *iPRES*, 2016.

[Med] *File types supported by Windows Media Player.* `https://support.microsoft.com/en-ca/help/316992/file-types-supported-by-windows-media-player`, vierailtu 2019-04-23.

[MET] *METS: An Overview Tutorial: Metadata Encoding and Transmission Standard (METS) OfficialWeb Site.* `https://www.loc.gov/standards/mets/METSOverview.v2.html`, vierailtu 2018-10-10.

[MP3] *Can I use... Support tables for HTML5, CSS3, etc.* `https://caniuse.com/mp3`, vierailtu 2019-01-02.

[OAS] *OASIS Open Document Format for Office Applications (OpenDocument) TC | OASIS.* `https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=office`, vierailtu 2018-11-26.

[PASa] *Sanasto | Digitalpreservation.fi.* `http://digitalpreservation.fi/specifications/sanasto`, vierailtu 2018-10-19.

[PASb] *Säilytys- ja siirtokelpoiset tiedostomuodot - v1.6.1.* `http://digitalpreservation.fi/files/PAS-tiedostomuodot-1.6.1.pdf`, vierailtu 2018-10-12.

[PDF] *mozilla/pdf.js: PDF Reader in JavaScript.* `https://github.com/mozilla/pdf.js/`, vierailtu 2019-01-09.

[Pho] *Photoshop image essentials.* `https://helpx.adobe.com/photoshop/using/image-essentials.html`, vierailtu 2019-02-22.

[PMC13] Palmer, William, May, Peter ja Cliff, Peter: *An Analysis of Contemporary JPEG2000 Codecs for Image Format Migration.* Teoksessa *10th International Conference on Preservation of Digital Objects*, sivu 197, 2013.

[PNGa] *PNG Frequently Asked Questions*. http://www.libpng.org/pub/png/pngfaq.html#patents, vierailtu 2019-04-23.

[Pngb] *PNG Suite Test Icons*. http://www.libpng.org/pub/png/pngsuite.html, vierailtu 2019-04-23.

[PNGc] *Portable Network Graphics (PNG) Specification (Second Edition)*. http://www.libpng.org/pub/png/spec/iso/index-object.html, vierailtu 2019-04-23.

[Pop] *Poppler*. https://poppler.freedesktop.org/, vierailtu 2019-01-09.

[RRG+00] Ritter, Niles, Ruth, Mike, Grissom, Brett Borup, Galang, George, Haller, John, Stephenson, Gary, Covington, Steve, Nagy, Tim, Moyers, Jamie, Stickley, Jim *et al.*: *Geotiff format specification geotiff revision 1.0*. SPOT Image Corp, 2000.

[RVW08] Rog, Judith ja Van Wijk, Caroline: *Evaluating file formats for long-term preservation*. Data Analysis and Knowledge Discovery, 24(1):83–90, 2008. https://www.kb.nl/sites/default/files/docs/KB_file_format_evaluation_method_27022008.pdf.

[RW16] Rimkus, Kyle R ja Witmer, Scott D: *Identifying Barriers To File Rendering In Bit-level Preservation Repositories: A Preliminary Approach*. International Conference on Digital Preservation (iPres) 2016, 2016.

[SCE01] Skodras, Athanassios, Christopoulos, Charilaos ja Ebrahimi, Touradj: *The JPEG 2000 still image compression standard*. IEEE Signal processing magazine, 18(5):36–58, 2001. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.313.4895&rep=rep1&type=pdf.

[Sig] *ODF: The Open Document Format | The Signal*. https://blogs.loc.gov/thesignal/2016/01/odf-the-open-document-format/, vierailtu 2019-01-09.

[Sus] *Sustainability of Digital Formats: Planning for Library of Congress Collections*. https://www.loc.gov/preservation/digital/formats/index.html, vierailtu 2019-04-19.

[TIF] *TIFF 6.0 Specification.* https://www.adobe.io/content/dam/udp/en/open/standards/tiff/TIFF6.pdf, vierailtu 2019-01-14.

[Toe15] Toevs, B.: *Processing of Metadata on Multimedia Using ExifTool: A Programming Approach in Python.* Teoksessa *2015 Annual Global Online Conference on Information and Computer Technology (GOCICT)*, sivut 26–30, Nov 2015.

[Ubu] *FreeFormats - Community Help Wiki.* https://help.ubuntu.com/community/FreeFormats?rev=36, vierailtu 2019-01-02.

[Usi] *FLAC - links.* https://xiph.org/flac/links.html#software, vierailtu 2019-02-01.

[VdK11] Knijff, Johan Van der: *JPEG 2000 for Long-term Preservation: JP2 as a Preservation Format.* D-Lib Magazine, 17(5):4, 2011. http://www.dlib.org/dlib/may11/vanderknijff/05vanderknijff.html, vierailtu 2019-02-20.

[Wal92] Wallace, Gregory K: *The JPEG still picture compression standard.* IEEE transactions on consumer electronics, 38(1):xviii–xxxiv, 1992. https://web.stanford.edu/class/ee398a/handouts/papers/Wallace%20-%20JPEG%20-%201992.pdf.

[WAV] *Wave File Specifications.* http://www-mmsp.ece.mcgill.ca/Documents/AudioFormats/WAVE/WAVE.html, vierailtu 2018-12-30.

[Web] *web-platform-tests/wpt: Test suites for Web platform specs — including WHATWG, W3C, and others.* https://github.com/web-platform-tests/wpt, vierailtu 2019-01-07.

[Wil12] Wilk, A.: *Patentability of Software.* Teoksessa *2012 IEEE International Conference on Software Science, Technology and Engineering*, sivut 30–39, June 2012.

A Tarkistuslista

Tämä tarkistuslista auttaa arvioimaan tiedostomuodon soveltuvuutta pitkäaikaissäilytykseen. Tarkistuslista koostuu tiedostomuotoa koostuvista piirteistä ja niille annettavista arvosanoista (hyvä, kohtalainen ja huono).

Tarkistuslista koostuu tärkeysjärjestykseen lajitelluista kysymyksistä, alkaen tärkeimpiä piirteitä koskevista kysymyksistä ja päättyen vähiten tärkeimpiin kysymyksiin. Tämä edesauttaa tiedostomuodon nopeampaa arviointia: jos tiedostomuodolle ei ole saatavilla standardia tai avoimen lähdekoodin ohjelmistoa, tiedostomuoto voidaan hylätä jo tässä vaiheessa.

Tarkistuslista ei määrittele tarkkoja ehtoja tiedostomuodon hylkäämiselle tai hyväksymiselle. Kysymykset toimivat suosituksina, joita käyttäen voidaan tarkastella tiedostomuodon soveltuvuutta säilytykseen.

Avoimen lähdekoodin ohjelmistojen saatavuus

☺	Useita avoimen lähdekoodin ohjelmistoja. Ohjelmistot tukevat tiedostomuodon lukemista ja kirjoittamista (esimerkiksi muuntaminen toiseen muotoon/muuntaminen toisesta tiedostomuodosta).
😐	Ainakin yksi avoimen lähdekoodin ohjelmisto. Ohjelmisto tukee ainakin tiedostomuodon lukemista.
☹	Ei avoimen lähdekoodin ohjelmistoja.

Avoimen lähdekoodin ohjelmiston kehitys

☺	Avoimen lähdekoodin ohjelmisto on aktiivisen kehityksen alla. Bugiraportteja käsitellään aktiivisesti. Ohjelmiston kehitystä tukee usea organisaatio.
😐	Avoimen lähdekoodin ohjelmisto on satunnaisen kehityksen alla. Bugiraporttien käsittely saattaa kestää pitkän aikaa. Ohjelmiston kehitystä tukee yksi organisaatio.
☹	Avoimen lähdekoodin ohjelmisto ei ole kehityksen alla. Ohjelmiston kehitystä ei tue yksikään organisaatio.

Digitaaliset kopionsuojausmenetelmät

	Tiedostomuoto ei tue DRM (Digital Rights Management, digitaalinen käyttöoikeuksien hallinta) -menetelmiä.
	Tiedostomuoto tukee vapaaehtoisia DRM (Digital Rights Management, digitaalinen käyttöoikeuksien hallinta) -menetelmiä.
	Tiedostomuoto käyttää DRM (Digital Rights Management, digitaalinen käyttöoikeuksien hallinta) -menetelmiä.

Standardointi

	Tiedostomuodosta on olemassa yksiselitteinen standardi. Dynaamisten tiedostomuotojen tapauksessa sallittuja tietueita voi rajata profiilien avulla. Tiedostomuodolle on olemassa tiedostoista koostuva testikokoelma, joilla ohjelmistojen yhteensopivuus voidaan tarkistaa.
	Tiedostomuotoa ei ole standardoitu. Tiedostomuodon käsittelyssä ei ole sovelluskohtaisia eroja
	Tiedostomuotoa ei ole standardoitu. Tiedostomuodon käsittelyssä on sovelluskohtaisia eroja.

Aineiston määrä ja muunnettavuus

Pitkäaikaissäilytykseen lähettävällä taholla saattaa olla hallussaan runsaasti säilytettävää aineistoa, joka ei täytä kaikkia säilytyksessä toivottuja piirteitä. Nämä aineistot voi tiedostomuodosta riippuen olla helposti muunnettavissa palvelun hyväksymään tiedostomuotoon tai vaatia ylimääräistä kehitystyötä muunnoksen toteuttamiseksi. Näiden kahden piirteen ollessa ristiriidassa suositaan ensisijaisesti aineiston säilyttämistä. Aineiston muuntaminen sopivampaan tiedostomuotoon hoidetaan myöhemmin.

	Paljon aineistoa	Jonkin verran aineistoa	Vähän aineistoa
Muunnettavissa helposti			
Muunnettavissa kohtuullisessa ajassa			
Vaikeasti muunnettavissa			

Metatietotuki

	Tiedostomuoto sisältää metatietotuen. Metatieto voi sisältää teknisiä piirteitä ja mahdollistaa säilytyksen eri työvaiheiden kuvaamisen.
	Tiedostomuoto sisältää metatietotuen. Metatieto voi sisältää aineistoa koskevia teknisiä piirteitä.
	Tiedostomuoto ei sisällä metatietotukea.

Tunnistettavuus

	Tiedostomuoto ja sen mahdollinen versionumero on tunnistettavissa helposti esimerkiksi tiedoston otsakkeessa sijaitsevan sarjan perusteella.
	Tiedostomuoto on tunnistettavissa helposti esimerkiksi tiedoston otsakkeessa sijaitsevan sarjan perusteella. Tiedostomuodon tarkka versionumero sen sijaan vaatii tiedoston rakennetta tulkitsevan ohjelmiston.
	Tiedostomuoto ei ole tunnistettavissa helposti ilman tiedoston rakennetta tulkitsevaa ohjelmistoa.

Pakkausmenetelmän häviöttömyys/häviöllisyys

Tiedoston pakkaaminen voi olla tarpeen jos tiedosto vaatii runsaasti tallennustilaa ja mahdolliset pakkausmenetelmät sallivat tiedoston tehokkaan pakkaamisen. Esimerkiksi kuvatiedostot on mahdollista pakata tehokkaammin vartavasten luotua pakkausmenetelmää käyttäen. Pakkaaminen ei välttämättä ole tarpeellista jos aineisto vie vain vähän tallennustilaa tai jos tiedoston pakkaamisesta saatavat säästöt ovat pienet.

	Tiedostomuoto tukee häviötöntä pakkausmenetelmää.
	Tiedostomuoto ei vaadi pakkausmenetelmän käyttöä.
	Tiedostomuoto tukee vain häviöllistä pakkausmenetelmää.

Ohjelmistopatentit

	Tiedostomuoto tai sen käsittelyssä käytetyt ohjelmistot eivät ole ohjelmistopatenttien alaisia.
	Tiedostomuoto tai sen käsittelyssä käytetyt ohjelmistot ovat ohjelmistopatenttien alaisia. Ohjelmistopatentit eivät estä tiedostomuodon käyttöä (esimerkiksi ohjelmistopatentti ei ole voimassa palvelua hallinnoivan yrityksen valtiossa).
	Tiedostomuoto tai sen käsittelyssä käytetyt ohjelmistot ovat ohjelmistopatenttien alaisia. Ohjelmistopatentit voivat haitata tiedostomuodon käyttöä.